

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra kybernetiky a biomedicínského inženýrství

# **Moderní způsoby ovládání energeticky nezávislých měřicích zařízení**

## **Modern methods of controlling energy independent measuring devices**

## Zadání diplomové práce

Student:

**Bc. Tereza Paterová**

Studijní program:

N2649 Elektrotechnika

Studijní obor:

3901T009 Biomedicínské inženýrství

Téma:

Moderní způsoby ovládání energeticky nezávislých měřicích zařízení  
Modern Methods of Controlling Energy Independent Measuring Devices

Jazyk vypracování:

čeština

Zásady pro vypracování:

### Teoretická část práce:

1. Popis energeticky nezávislých měřicích zařízení, jejich aplikačních oblastí a zpracování přehledu centrálních řídicích jednotek (MPU, MCU, FPGA apod.).
2. Rešerše soft-computing metod vhodných pro nízkourovňovou implementaci.

### Praktická část práce:

1. Zpracování analýzy implementovatelnosti vybrané metody a implementace této metody v simulačním frameworku v jazyku C#.
2. Testování implementované metody, export výstupních dat a statistické zpracování výsledků simulace.
3. Zhodnocení výsledků práce a navržení budoucích postupů řešení.

Seznam doporučené odborné literatury:

- [1] CHATURVEDI, Devendra K. *Soft computing techniques and its applications in electrical engineering*. Berlin: Springer, 2008. ISBN 978-3540774815.
- [2] MURPHY, Kevin P. *Machine learning: a probabilistic perspective*. Cambridge, MA: MIT Press, c2012. ISBN 978-0262018029.
- [3] SUTTON, Richard S. a Andrew G. BARTO. *Reinforcement learning: an introduction*. Cambridge, Mass.: MIT Press, c1998. ISBN 978-0262193986.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Michal Prauzek, Ph.D.**

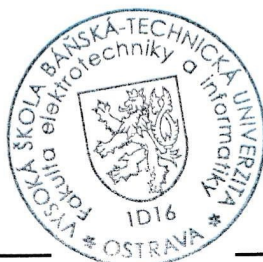
Konzultant diplomové práce: doc. Ing. Marek Penhaker, Ph.D.

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2020



doc. Ing. Jiří Koziorek, Ph.D.  
vedoucí katedry

  
prof. Ing. Pavel Brandštetter, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně. Uvedla jsem všechny literární  
prameny a publikace, ze kterých jsem čerpala.

V Ostravě 30. dubna 2020

Tereza Polková

Ráda bych na tomto místě poděkovala panu doc. Ing. Michalovi Prauzkovi, Ph.D. a panu Ing. Jaromírovi Konečnému, Ph.D. za všechny odborné rady, ochotu a trpělivost, kterou mně při vedení diplomové práce oba poskytovali.

## Abstrakt

Cílem této práce je najít optimální strategii pro četnost měření energeticky nezávislého měřicího zařízení. Toto zařízení je umístěno v prostředí bez přístupu k elektrické síti. V práci jsou popsány jednotlivé soft-computing metody, které by se pro řešení této problematiky daly použít. Výsledný algoritmus využívá zpětnovazebního učení, konkrétně Q-learning. Chování algoritmu bylo otestováno ve vytvořené simulaci v programovacím jazyku *C#.NETCore*. Predikované hodnoty byly vypočítávány pomocí aproximace polynomem. Pro testování simulace byla použita meteorologická data v průběhu pěti let a šest kombinací konstant rychlosti učení a stupně polynomu. Výsledkem testování je, že kvalitu predikce ovlivňuje stupeň polynomu.

**Klíčová slova:** Soft-computing metody, Q-learning, Energeticky nezávislá měřicí zařízení, Aproximace polynomem

## Abstract

The aim of this thesis is to find an optimal strategy for the frequency of measurement of energy independent measuring device. This device is located in an environment, where it is not possible to supply energy demands from power grid. The thesis describes particular soft-computing methods that could be used to solve this problem. The final algorithm uses reinforcement learning, specifically Q-learning. The behavior of the algorithm was tested in the designed simulation. The simulation was programmed in *C#.NETCore*. The predicted values were calculated using polynomial approximation. Meteorological data over five years and six combinations of learning rate and polynomial degree were used to test the simulation. The testing shows that the quality of the prediction is influenced by the degree of the polynomial.

**Key Words:** Soft-computing methods, Q-learning, Energy independent measuring device, Polynomial approximation

# Obsah

<b>Seznam obrázků</b>	<b>7</b>
<b>Seznam tabulek</b>	<b>8</b>
<b>1 Úvod</b>	<b>9</b>
<b>2 Energeticky nezávislá měřicí zařízení</b>	<b>10</b>
2.1 Sběr energie . . . . .	10
2.2 Centrální řídicí jednotky . . . . .	12
2.3 Aplikační oblasti . . . . .	14
<b>3 Soft-computing metody</b>	<b>16</b>
3.1 Neuronové sítě . . . . .	16
3.2 Fuzzy logika . . . . .	18
3.3 Q-learning . . . . .	20
<b>4 Návrh simulačního softwaru</b>	<b>23</b>
4.1 Zdrojová data pro simulaci . . . . .	24
4.2 Návrh Q-learning kontroléru . . . . .	24
4.3 Implementace Q-learning kontroléru . . . . .	31
<b>5 Metrika vyhodnocení</b>	<b>33</b>
5.1 Párové testy . . . . .	33
5.2 MSE, RMSE a SD . . . . .	34
5.3 Kvartilová analýza . . . . .	35
5.4 Analýza četnosti . . . . .	35
<b>6 Testování a vyhodnocení výsledků</b>	<b>36</b>
6.1 Testování akcí Q-learningu – Perioda $T$ . . . . .	36
6.2 Testování stavů Q-learningu – Relativní chyba predikce $\varepsilon_{rel}$ . . . . .	38
6.3 Testování odměňování Q-learningu . . . . .	40
6.4 Testování přesnosti predikce . . . . .	41
6.5 Vyhodnocení výsledků testování . . . . .	50
<b>7 Závěr</b>	<b>51</b>
<b>Literatura</b>	<b>53</b>
<b>Přílohy</b>	<b>55</b>

## Seznam obrázků

1	Topologie systému pro sběr energie . . . . .	11
2	Model neuronu . . . . .	16
3	Model perceptronu . . . . .	17
4	Vrstva neuronové sítě . . . . .	17
5	Aplikace fuzzy logiky . . . . .	18
6	Markovův rozhodovací proces . . . . .	21
7	Blokové schéma návrhu simulačního softwaru . . . . .	23
8	Blokové schéma simulačního frameworku . . . . .	23
9	Blokové schéma návrhu Q-learning kontroléru . . . . .	24
10	Blok Buffer . . . . .	25
11	Princip FiFoBufferu . . . . .	25
12	Graf lineárního polynomu . . . . .	26
13	Graf polynomu 2. stupně . . . . .	27
14	Graf polynomu 5. stupně . . . . .	27
15	Blok $\varepsilon_{rel}$ . . . . .	28
16	Blok Q-learning se sadou pravidel . . . . .	28
17	Graf závislosti časového úseku $T$ na odměně . . . . .	29
18	Graf závislosti relativní chyby predikce $\varepsilon_{rel}$ na odměně . . . . .	30
19	Histogram periody $T$ v dané kombinaci konstant . . . . .	37
20	Histogram periody $T$ v dané kombinaci konstant . . . . .	38
21	Histogram relativní chyby predikce $\varepsilon_{rel}$ v dané kombinaci konstant . . . . .	39
22	Histogram relativní chyby predikce $\varepsilon_{rel}$ v dané kombinaci konstant . . . . .	39
23	Histogram odměn v dané kombinaci konstant . . . . .	40
24	Histogram odměn v dané kombinaci konstant . . . . .	41
25	Q-Q graf odchylek $X_m$ a $X_{predict}$ při kombinacích konstant $\alpha = 0,5$ $N = 1$ . . .	43
26	Boxplot odchylek $X_m$ a $X_{predict}$ při jednotlivých kombinacích konstant . . . . .	44
27	Graf srovnání průběhů solární energie v závislosti na čase u $\alpha = 0,2$ $N = 1$ . . .	46
28	Graf srovnání průběhů solární energie v závislosti na čase u $\alpha = 0,2$ $N = 1$ . . .	47
29	Graf srovnání průběhů solární energie v závislosti na čase u $\alpha = 0,2$ $N = 1$ . . .	47
30	Graf srovnání průběhů solární energie v závislosti na čase u $\alpha = 0,2$ $N = 1$ . . .	47
31	Grafy srovnání průběhů solární energie v závislosti na čase u $\alpha = 0,2$ $N = 1$ . . .	49

## Seznam tabulek

1	Interpretace logických spojek . . . . .	19
2	Ukázka zdrojových dat . . . . .	24
3	Hodnoty jednotlivých akcí (periody $T$ ) a jejich odměny . . . . .	30
4	Hodnoty jednotlivých stavů (relativní chyba predikce $\varepsilon_{rel}$ ) a jejich odměny . . . . .	31
5	Kombinace jednotlivých expertních konstant a stupně polynomu . . . . .	36
6	Celkový počet měření při jednotlivých kombinacích konstant . . . . .	36
7	Absolutní (relativní) četnost period $T$ při jednotlivých kombinacích konstant . . . . .	36
8	Absolutní (relativní) četnost period $T$ při jednotlivých kombinacích konstant . . . . .	37
9	Hodnoty kvartilů periody $T$ při jednotlivých kombinacích konstant . . . . .	38
10	Hodnoty kvartilů relativní chyby predikce $\varepsilon_{rel}$ . . . . .	38
11	Hodnoty kvartilů odměn/penalizace za akci a stav . . . . .	40
12	Hodnoty kvartilů přejatých dat $X_m$ a predikovaných dat $X_{predict}$ . . . . .	42
13	Hodnoty rozdílů průměru $X_m$ a $X_{predict}$ . . . . .	42
14	Výsledky párového porovnávání odchylek $X_m$ a $X_{predict}$ . . . . .	44
15	Výsledky MSE, RMSE a SD odchylek $X_m$ a $X_{predict}$ . . . . .	45
16	Procentuální vyjádření úbytku dat při použití simulace oproti původním $X_m$ . . . . .	46



# 1 Úvod

Energeticky nezávislá měřicí zařízení jsou hojně využívána v různých oblastech techniky – monitorování životního prostředí, varovné systémy před přírodními riziky, nositelná elektronika a mikrosystémy monitorující lidské zdraví. Jejich princip tkví v tom, že nemají přístup k elektrické síti a nejsou ovlivněny přítomností člověka. Cílem této práce je vytvořit algoritmus schopný řízení energeticky nezávislého systému s efektivním měřením dat.

Nejprve (viz kapitola č. 2) je představeno energeticky nezávislé zařízení jako takové. Jsou popsány oblasti, ve kterých nalézá využití. Zmíněny jsou také jeho jednotlivé součásti, zejména centrální řídicí jednotka a systém pro sběr energie.

V další části práce, kapitola č. 3, jsou popsány softcomputing metody, které se hodí k řízení energeticky nezávislých zařízení – neuronové sítě, fuzzy logika a zpětnovazební učení. Tyto metody mají společnou vlastnost a to, že se hodí pro nízkou úrovní implementaci, což je jeden z klíčových požadavků energeticky nezávislého zařízení.

Pro implementaci byl vybrán algoritmus zpětnovazebního učení, Q-learning. Tento algoritmus je v této oblasti používán, jelikož byla prokázána schopnost nalézt optimální řešení pro každý konečný Markovův rozhodovací proces [2]. A také je tento algoritmus vhodný pro učení simulací [1].

Byl vytvořen řídicí algoritmus, který řídí činnost simulace, viz kapitola č. 4. Algoritmus je složen z několika částí – první slouží pro ukládání dat a výpočet predikované hodnoty, druhá pro výpočet relativní chyby predikce a třetí obsahuje samotný Q-learning se sadou pravidel. Algoritmus byl naučen předpovídat informace o sluneční energii a na základě toho řídil frekvenci měření, aby nedocházelo k redundantnímu měření a zároveň, aby byla naměřena všechna důležitá data. Pro učení byla použita meteorologická data z průběhu pěti let, která byla měřena s pěti minutovými intervaly.

Chování algoritmu bylo otestováno ve simulaci vytvořené v programovacím jazyku *C#.NET Core*. V kapitole č. 6 byly srovnány výsledky simulace s výchozími daty a statisticky vyhodnoceny. Ke statistickému vyhodnocení byly použity metriky – párové testování, výpočet MSE, RMSE a SD, kvartilová analýza a analýza četností (viz kapitola č. 5).

## 2 Energeticky nezávislá měřicí zařízení

Energeticky nezávislá měřicí zařízení jsou přístroje, které byly navrženy pro sběr dat na vzdálených místech, kde jsou omezené možnosti údržby. Je proto nezbytné, aby systém efektivně využíval dostupnou energii. Provoz systému, jmenovitě získávání environmentálních dat a následný přenos dat do sítě, je nutno řídit inteligentním výpočetním systémem [8]. Tyto zařízení jsou umístěna na vzdálených místech bez přístupu k elektrické síti nebo možnosti zásahu člověka. Proto existuje značná potřeba využití pokročilých řídicích strategií pro dosažení spolehlivé funkčnosti [7].

Systémy vzdáleného monitorování prostředí vyžadují sofistikované řízení spotřeby energie k dosažení dlouhodobého provozu s minimální údržbou. Existuje několik způsobů, jakým je energie poskytována a používána a tím se určuje, jak dlouho může zařízení fungovat, aniž by bylo nutné do něj zasahovat. Může být řízeno řadou technik, obvykle kombinovaných do strategií řízení energie. Cílem je minimalizovat spotřebu energie, při zachování kvality služby na žádoucí úrovni [14].

Měřicí systém, který je energeticky nezávislý je složen z hardwarových obvodů spolu s algoritmem pro řešení dané problematiky. Mezi hardwarové komponenty, které umožňují funkci a komunikaci s okolím, se řadí zdroj, řídicí jednotka, paměť, časovače, komunikační porty a výstupní obvody. Systém řízení je ovládán mikroprocesorem (MPU), mikrokontrolérem (MCU), systémem na čipu (SoC) či programovatelným hradlovým polem (FPGA). Tato zařízení pro svou činnost nepotřebují trvalé ani krátkodobé připojení k elektrické síti [13].

### 2.1 Sběr energie

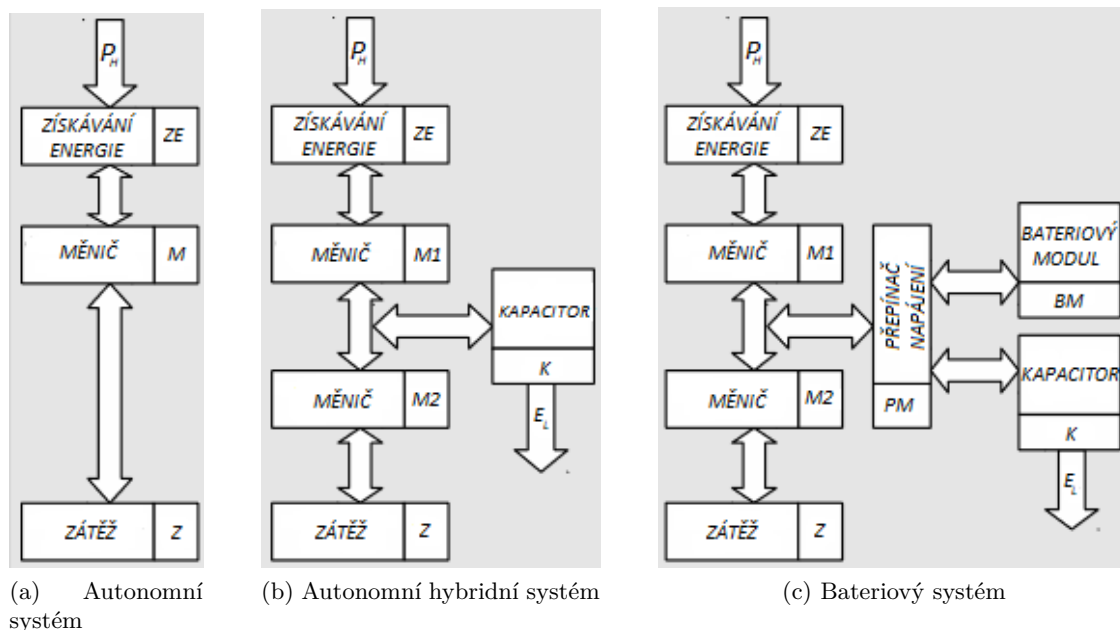
Vzhledem k tomu, že energeticky nezávislá zařízení musí splňovat podmínku energetické neutrality, tak pro systém sběru energie byly vytvořeny tři hlavní topologie (viz. obr. 1):

- autonomní,
- autonomní hybridní,
- bateriová.

V závislosti na konfiguraci a cíli návrhu je vybrána strategie řízení energie [14]. Takové topologie hardwaru představují výzvu k implementaci inteligentních řídicích strategií, kde jádro problému spočívá ve vyrovnávání intervalu měření a přenosu dat [7].

#### 2.1.1 Autonomní systém

Autonomní systémy sběru energie plně uspokojují své energetické potřeby z okolních zdrojů, bez využití baterií. Struktura autonomního sklízecího systému je znázorněna na obr. 1a. Mohou fungovat pouze tehdy, je-li k dispozici zdroj energie v prostředí. Jejich životnost a výkon nejsou omezeny vlastnostmi energetického uložště, například samovybíjení, omezený počet cyklů a



Obrázek 1: Topologie systému pro sběr energie [14]

stárnutí uložště. Tyto systémy musí splňovat tzv. energetickou neutralitu, protože nemohou nikdy spotřebovat více energie, než mohou z prostředí získat. Musí být navrženy na maximální výkon, tj. aby fungovaly na maximální úrovni, která může být podporována v daném prostředí. Pro podporu těchto cílů návrhu by autonomní systémy měly používat predikční algoritmy, které poskytují odhady budoucí dostupné energie v průběhu času. Správná strategie řízení energie by měla takovému systému umožnit dosáhnout požadovaného využití v prostředí s proměnnou energií v závislosti na čase [14]. Nevýhodou této topologie je, že pokud spotřebič (Z) neodebere získanou energii, přebytečná energie bude ztracena, jelikož modul pro získávání energie (ZE) je čistý zdroj a není zde možnost ukládání energie. Měníčem (M) je většinou DC/DC měnič, který přímo napájí spotřebič (Z). Výhodou tohoto modelu je využití pouze jednoho modulu pro konverzi energie (M), což ovlivňuje celkovou výkonovou ztrátu systému [21].

### 2.1.2 Autonomní hybridní systém

Autonomní hybridní systém je nejčastějším typem energetických systémů. Struktura systému je znázorněna na obr. 1b. Tato topologie využívá k napájení libovolný zdroj energie z prostředí. Oproti autonomnímu systému mají implementovaný zásobník energie (K) ve formě sekundární baterie nebo superkapacitoru. Zařízení shromažďuje energii pro provoz systému a dobíjení zásobníku. Toto uspořádání může výrazně prodloužit provozní životnost systému. Se správnou metodou řízení energie může tato topologie dosáhnout 100 % provozu bez výpadku z důvodu vyčerpání energie. Sekundární baterie a zařízení pro odběr energie musí být dimenzovány tak, aby uspokojovaly energetické potřeby systému a splňovaly princip energetické neutrality. Systém

může někdy spotřebovávat více energie, než poskytuje zdroj, ale míra produkce/spotřeby musí být dlouhodobě vyrovnaná. Nevýhodou je samovybíjení energetického zásobníku, což je dáno konstrukcí sekundární baterie či kapacitoru [21, 14].

### 2.1.3 Bateriový systém

Tyto systémy mají obvykle baterii jako hlavní zdroj energie a zařízení pro sběr energie, které hraje důležitou, ale sekundární roli. Cílem správy energie v takových systémech je omezit spotřebu energie z baterií a prodloužit životnost systému, například snížením frekvence externího nabíjení nebo výměny baterií. Tento systém může používat primární nebo sekundární baterie. Struktura bateriového sklízecího systému je znázorněna na obr. 1c. Získaná energie z prostředí může přímo nebo nepřímo pohánět zátěž nebo její specifické části. Tento přístup výrazně zvyšuje spolehlivost systému a umožňuje sběr, zpracování nebo přenos dat. Dokud zbývá energie v primární baterii, systém může pokračovat v činnosti v situacích, kdy je vyčerpán sekundární zdroj a není k dispozici získání environmentální energie.

Tato topologie obsahuje, oproti předchozí, navíc primární bateriovou část (BM) a multiplexor pro přepínání mezi zdroji energie (PM) [21].

## 2.2 Centrální řídicí jednotky

Centrální řídicí jednotky jsou určeny pro zpracování dat a koordinaci časových, ukládacích a přenosových operací systému. Řídicí jednotka se používá k řízení adaptivních cyklů činnosti systému [15]. Řadíme mezi ně mikrokontrolér (MCU), mikroprocesorovou jednotku (MPU), programovatelná hradlová pole (FPGA) a systém na čipu (SoC).

### 2.2.1 MCU

Mikrokontrolér (MCU) je jednočip, primárně určen ke kontrole a řízení jiných elektronických zařízení. Jedná se o kompaktní integrovaný obvod určený k řízení konkrétní operace ve vestavěném systému [15].

MCU se nacházejí mimo jiné ve vozidlech, robotech, lékařských zařízeních, prodejních automatech a domácích spotřebičích. Jsou to v podstatě jednoduché miniaturní zařízení určené k ovládání funkcí větší komponenty, bez složitého operačního systému. Mikrokontroléry spotřebovávají jen velmi málo energie, mohou běžet relativně pomalu a provádět jednotlivé programy specifické pro daný úkol [15]. Tuto energetickou přednost si udržují pomocí pokročilého energetického managementu [21].

Základní prvky mikrokontroléru:

- CPU – Procesor lze považovat za mozek zařízení. Zpracovává a odpovídá na různé pokyny, které řídí funkci mikrokontroléru. To zahrnuje provedení základních aritmetických, logických a I/O operací.

- Programová paměť – Ukládá dlouhodobé informace o instrukcích, které CPU provádí.
- Datová paměť – Je vyžadována pro dočasné uložení dat během provádění instrukcí.
- I/O periferie – Vstupní a výstupní zařízení jsou rozhraním procesoru pro vnější svět. Vstupní porty přijímají informace a odesílají je do procesoru ve formě binárních dat. Procesor přijímá tato data a odešle potřebné instrukce na výstupní zařízení, která vykonávají úkoly mimo mikrokontrolér [15].

Architektura mikrokontroléru může být založena na architektuře Harvard nebo Von Neumann, což nabízí různé způsoby výměny dat mezi procesorem a pamětí. S architekturou Harvard jsou datová sběrnice a instrukce oddělené, což umožňuje současné přenosy. V architektuře Von Neumann je jedna sběrnice použita jak pro data, tak pro instrukce [15].

Tyto výkonné jednočipy jsou vhodné pro rozsáhlejší aplikace metod řízení – např. softcomputing metody a pokročilé zpracování signálů [21].

### 2.2.2 MPU

Mikroprocesorová jednotka (MPU) je zařízení, které implementuje základní prvky počítačového systému na jeden integrovaný obvod, nebo několik integrovaných obvodů fungujících jako soudržná jednotka, určené pro zpracování digitálních dat.

Je to programovatelné zařízení, které vstupuje, provádí nad ním aritmetické a logické operace a vytváří požadovaný výstup. Zjednodušeně řečeno, mikroprocesor je digitální zařízení na čipu, které dokáže načíst instrukce z paměti, dekodovat a provést je a dát výsledky.

Moderní MPU obvykle zahrnují funkčnost hodin, centrální procesorové jednotky (CPU), aritmetické logické jednotky (ALU), jednotky s pohyblivou řádovou čárkou (FPU), řídicí jednotky (CU), jednotky pro správu paměti (MMU), přerušení, rozhraní vstupu/výstupu a mezipaměť. Specializovaný mikroprocesor může také sloužit jako jednotky grafického zpracování (GPU), jednotky zpracování signálu (DSP), jednotky neurálního zpracování (NPU) atd. [16].

### 2.2.3 FPGA

FPGA je zkratka pro programovatelné hradlové pole. Jedná se o polovodičové zařízení, které obsahuje matici konfigurovatelných logických bloků připojených prostřednictvím programovatelných propojení. Obvykle obsahují stovky přizpůsobitelných I/O pinů. Tyto integrované integrované čipy obsahují řadu logických bran, nejčastěji NAND bran, které mohou být „programovány“ výrobcem k provedení řady specifických logických funkcí. Jak název napovídá, logika v FPGA může být naprogramována v zařízení polí, obvykle při spuštění systému. Toto programování lze čas od času aktualizovat, což umožňuje pohodlnou údržbu [17].

Oproti mikrokontroléru jsou FPGA odlišné. Uživatel je ten, kdo navrhuje obvod. Neexistuje žádný procesor, na kterém by se mohl software spouštět, alespoň dokud jej uživatel nenavrhne. Mírnou nevýhodou pro FPGA je, že ukládají svou konfiguraci v RAM, ne flash, což znamená, že

jakmile ztratí energii, ztratí svou konfiguraci. Musí být nakonfigurovány při každém zapnutí napájení [18].

FPGA poskytují výhody konstruktérům mnoha typů elektronických zařízení, od inteligentních energetických sítí, navigace letadel, asistence řidiče automobilu až po lékařské ultrazvuky [18].

#### 2.2.4 SoC

Systém na čipu (SoC) kombinuje požadované elektronické obvody různých počítačových komponent do jednoho integrovaného čipu. Jedná se o kompletní elektronický systém, který může obsahovat analogové, digitální nebo vysokofrekvenční obvody. Protože SoC zahrnuje hardware i software, používá méně energie, má lepší výkon, vyžaduje méně místa a je spolehlivější než multičipové systémy. Díky jejich malé spotřebě energie je jejich typické využití v oblasti vestavěných systémů [12].

SoC může mít řadu prvků v závislosti na účelu jeho využití. Zde je uveden seznam těch nejběžnějších:

- CPU – Centrální procesorová jednotka je jádrem SoC, která je zodpovědná za většinu výpočtů a rozhodnutí. Přijímá vstup z jiných hardwarových komponent a softwaru a poskytuje odpovídající výstupní odezvy. Většina procesorů má dnes dvě, čtyři nebo osm jader, aby bylo možné provádět paralelní zpracování.
- GPU – Jedná se o zkratku pro grafickou procesorovou jednotku, což je koprocesor na provádění a urychlení grafických operací.
- RAM – Operační paměť, jenž umožňuje spouštění procesů na čipu.
- ROM – Paměť pro uložení firmwaru a operačního systému.
- další – Regulátory napětí a obvody řízení spotřeby, externí rozhraní, periferie (hodiny, časovače, generátory), sběrnice periferních rozhraní a analogová rozhraní (A/D převodníky) aj. [12].

### 2.3 Aplikační oblasti

Mimo oblasti monitorování environmentálního prostředí se energeticky nezávislá měřicí zařízení využívají jako varovné systémy před přírodními riziky, u nositelné elektroniky a mikrosystémů, které monitorují fyziologickou veličinu – např. krevní tlak a hodnota glukózy v krvi. V neposlední řadě tento systém nalézá využití u vytváření tzv. e-skin.

Varovné systémy mohou být aplikovány v mnoha různých odvětvích. Mezi nejčastější se řadí detekce požárů ve volné přírodě, detekce výšky hladin toků, detekce sesuvu půdy a padajícího kamení.

Další aplikační oblastí je nositelná elektronika. Jedná se o moderní typ komerční elektroniky, která je navržena tak, aby mohla být běžně nošena člověkem, jelikož je upevněna přímo

na lidské tělo. Řadí se tam – chytré hodinky, špionážní technika, personální lékařská diagnostika, sportovní elektronika aj. Tato elektronika může být napájena zdroji z prostředí, např. termoelektrické zdroje, piezoelektrické měniče nebo může být jako zdroj použito lidské tělo [21].

Další oblastí jsou mikrosystémy, které umožňují nepřetržité monitorování nitroočního tlaku. Mikrosystém získává energii ze slunečního záření pomocí integrovaného solárního článku, který slouží k dobíjení baterie a napájení obvodů. Používá techniku obvodů s velmi nízkým výkonem k dosažení energetické autonomie, prodlužuje životnost na neurčitou dobu a potenciálně dává lékařům možnost dostat se na lepší úroveň u léčby glaukomu. Techniky energeticky autonomních obvodů jsou potřebné pro činnost mikrosystémů, které by mohly být použity jako součást budoucích lékařských implantátů. Například senzory krevního tlaku a glukózy [35].

Využití u vývoje autonomního senzoru glukózy je následovné. Byl vyvinut flexibilní, vysoce citlivý indikátor glukózy na bázi nanocelulózy. Tento systém je schopen detekovat glukózu a současně napájet indikátor (LED). Toto zařízení bylo postaveno na speciálně vyvinutých deskách s plošnými spoji z nanocelulózy. Z toho vychází, že indikátor je plně biokompatibilní. Výsledkem je tedy autonomní ukazatel glukózy. Velikost zařízení je  $0,5 \text{ cm}^2$  a hmotnost méně než 40 mg. Tento senzor může být použit v kombinaci s jinými aplikacemi pro monitorování lidského zdraví [37].

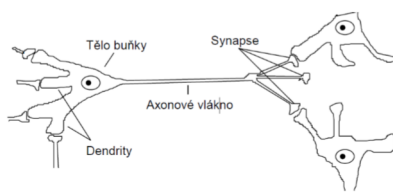
Další oblastí využití energeticky nezávislých systému je u vytváření systému e-skin (elektronická kůže). Elektronická kůže by měla být pružná samo léčivá elektronika, která je schopna napodobit funkce lidské nebo zvířecí kůže. Materiály v sobě obsahují snímače jejichž účelem je reprodukovat schopnosti stejné jako má lidská kůže –např. reagovat na faktory prostředí, jako jsou změny tepla a tlaku. Využití bude nalezeno v oblasti robotiky, protetiky a nositelné elektroniky [36].

### 3 Soft-computing metody

Jedná se o metody, u níž se výpočetní systémy „polidšťují“ a jsou učeny pracovat samostatně bez zásahu člověka. Schopností soft-computing metod je naučit se konkrétní úlohu z dat nebo pozorování. Jedná se o sadu adaptivních mechanismů, které umožňují systémům inteligentní chování ve složitých a měnících se prostředích. Tyto mechanismy zahrnují paradigmaty, které mají schopnost učit se, přizpůsobovat se novým situacím, zobecňovat, objevovat. Simulují biologickou inteligenci. Mezi tyto metody se řadí např. fuzzy logika, neuronové sítě a zpětnovazební učení. [3].

#### 3.1 Neuronové sítě

Systém založený na bázi jednoduchého fyziologického neuronu, jenž se skládá z několika dendritů, které reprezentují místo pro vstup signálu do těla neuronu. Tělo sčítá signály přicházející na jednotlivé dendrity. Poté dochází k excitaci neuronu a přenesení vzniklého napětí pomocí axonu na synapse, které tento signál upravují a přesunují na dendrity následujícího neuronu [5].



Obrázek 2: Model neuronu [5]

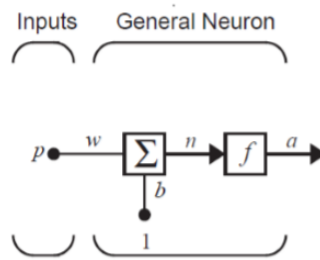
##### 3.1.1 Model perceptronu

Základem neuronových sítí je umělý neuron neboli perceptron. Je tvořen vstupní hodnotou  $p$ , která je následně vynásobena hodnotou váhy  $\omega$  daného perceptronu a přesunuta na sumátor. Tady dojde k přičtení hodnoty  $b$  která udává vnitřní potenciál perceptronu. Výstup sumátoru  $n$  je přiveden na příslušnou přenosovou funkci neuronu  $f()$  (viz obr. 5) a dochází k vypočtení výsledné hodnoty perceptronu  $a$ . Tento proces funkce perceptronu lze zapsat pomocí následující rovnice:

$$a = f(\omega p + b) \quad (1)$$

Hodnota vah  $\omega$  odpovídá síle synapsí u biologického neuronu. Sumátor a přenosová funkce  $f()$  reprezentuje tělo neuronu. Výstup  $a$  je pak napětí přenesené na axon biologického neuronu. Hodnoty vah a vnitřního potenciálu perceptronu se mohou upravovat. K jejich změnám dochází během procesu učení perceptronu, až do doby, než výstup neuronu bude odpovídat realitě [6].





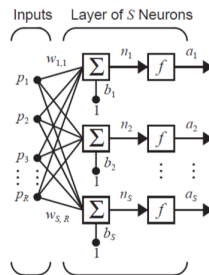
Obrázek 3: Model perceptronu [6]

### 3.1.2 Přenosová funkce

Podle toho, jaký problém bude pomocí perceptronu řešen, se volí vhodná přenosová funkce. Existuje celá řada lineárních a nelineárních přenosových funkcí  $f()$ . Mezi nejčastěji používanými jsou hardlimit, lineární a sigmoidální [6].

### 3.1.3 Vrstva neuronové sítě

Perceptron sám o sobě nemá dostatečné možnosti pro řešení složitějších problémů. V praxi se z toho důvodu používá velké množství vzájemně propojených perceptronů. Spojují se paralelně a vytváří tak vrstvy neuronové sítě (viz obr. č. 4). Každý vstup vrstvy  $p$  je spojen se všemi perceptrony tvořící vrstvu neuronů pomocí matice vah. Jednotlivé perceptrony mají vlastní hodnoty vnitřního potenciálu  $b$  a váhy  $w$ . Rozdílně se mohou volit i přenosové funkce perceptronů [6].



Obrázek 4: Neuronové sítě s jednou vrstvou [6]

Vrstvy neuronů lze dále sériově kombinovat, čím je vytvořena komplexní neuronová síť schopna vyřešit i náročné úlohy. V praxi se většina neuronových sítí skládá maximálně ze tří vrstev [6].

### 3.1.4 Využití neuronových sítí

Neuronové sítě mohou být využity při řešení celé řady aplikací. Prvním z příkladů je pro řešení problémů týkajících se aproximace funkcí. Pro trénování neuronové sítě je třeba použít datový

set skládající se ze závislých výstupních proměnných a jedné nebo více vstupních nezávislých proměnných. Sít si v průběhu trénovacího procesu vytvoří závislost mezi těmito proměnnými, která bude použita pro následující určování výstupu. Konkrétním příkladem využití aproximace funkcí je senzor detekce polohy objektu, kdy jsou vstupní proměnné napětí snímány pomocí dvou světelných senzorů a výstupem neuronové sítě je poloha objektu.

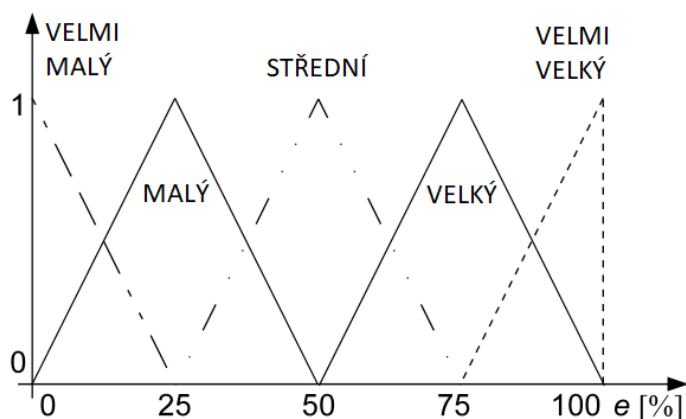
Další oblastí využití je rozpoznávání vzorků, jenž se používá, pokud chceme vstupní proměnné roztrždit do několika kategorií. Tyto neuronové sítě našli své využití například při rozpoznávání maligních a benigních nádorů.

Podobnou oblastí je shlukování, které se používá pro setřídění dat podle jejich vzájemné podobnosti. Neuronové sítě na bázi shlukování se používají pro analýzu velkého množství dat. Sít při trénování nemá určený výsledek a musí sama odhadnout požadovaný výstup podle charakteru vstupních dat.

Poslední oblastí je predikce, jenž má u neuronových sítí využití v komplexních dynamických systémech, kde je potřeba odhadnout budoucí hodnoty na základě předchozích výsledků [6].

### 3.2 Fuzzy logika

Klasická teorie množin umožňuje, aby prvky byly buď začleněny do množiny, nebo ne. To je v rozporu s lidským uvažováním, které zahrnuje míru nepřesnosti nebo nejistoty, které se vyznačuje použitím lingvistických proměnných, jako je většina, mnoho, často, zřídka atd. Toto uvažování je modelováno fuzzy logikou, která je vícehodnotová (viz obr. č. 5). To znamená, že například hodnota 20 % náleží do množiny VELMI MALÝ a zároveň MALÝ. Fuzzy množiny se liší od klasických množin tím, že umožňují, aby byl objekt částečným členem více množin [3].



Obrázek 5: Aplikace fuzzy logiky [10]

Výroková logika používá pouze dvě logické hodnoty – pravdu (1) a nepravdu (0). Fuzzy logika může nabývat všech hodnot intervalu  $\langle 0; 1 \rangle$ , kterých je nekonečně mnoho. Umožňuje vyjádřit částečnou příslušnost k množině. Je používán stupeň příslušnosti jako matematický

model vágnosti<sup>1</sup>, zatímco pravděpodobnost je matematický model neznalosti. Typické funkce příslušnosti používané ve fuzzy expertních systémech jsou trojúhelníky a lichoběžníky [3].

Fuzzy výstupy všech pravidel (určené pomocí stupně příslušnosti) jsou pak přepočítány za vzniku výstupního fuzzy setu [10].

### 3.2.1 Fuzzyfikace

Jedná se o operaci, která převádí reálné proměnné do jazykové podoby. Příkladem je velikost periody, což je reálná proměnná. Po fuzzyfikaci získá jazykovou podobu se třemi atributy – malá, střední, velká.

Musí být určen tzv. stupeň členství  $u$  v příslušných množinách. Tento parametr musí být určen, jak pro jazykovou proměnnou, tak pro další proměnnou např. četnost využití periody při dané velikosti, která je dále využita u defuzzifikace. Stupeň členství  $u$  nabývá hodnot od 0 do 1. Zařazení do skupiny je určeno funkcí, nejčastěji se používají funkce lambda, S-typ a Z-typ. Tyto funkce určují přiřazení proměnné a jejich atributů k dané množině [4].

### 3.2.2 Fuzzy inference

Fuzzy interferencí jsou definována transformační pravidla. Tato pravidla modelují situaci pomocí podmínkové věty IF *podmínka* THEN *proved*. Tyto podmínky mohou být i složené, tzn., že jsou spojené logickými spojkami (viz tabulka č. 1). Příkladem je IF *perioda = malá* THEN *malá četnost* [4].

Tabulka 1: Interpretace logických spojek [4]

AND	$u_{xANDy} = \min(u_x, u_y)$
OR	$u_{xORy} = \max(u_x, u_y)$
NOT	$u_{NOTx} = 1 - u_x$

### 3.2.3 Defuzzyfikace

V této části probíhá opačný postup oproti fuzzyfikaci – převod jazykových proměnných do reálných proměnných. Využívá se různých transformačních postupů. Příkladem je princip maxima nebo těžiště výsledné fuzzy množiny.

Pro principu maxima charakteristické funkce množiny platí hodnota pro celý interval řízené proměnné. Jako defuzzikovaná hodnota se poté bere střed intervalu.

U principu těžiště je defuzzikovaná hodnota těžiště výsledné fuzzy množiny [4].

---

<sup>1</sup>subjektivní složka hodnocení [4]

### 3.2.4 Využití fuzzy logiky

Fuzzy množiny jsou používány k popisu pravděpodobných, nejistých nebo neurčitých jevů, s nimiž se pracuje. Mezi příznivé vlastnosti fuzzy logiky patří snadnost návrhu a implementace, v kombinaci se schopností zpracovat kompletní a nepřesná data. Proto byla úspěšně použita v řídicích systémech (např. řízení subsystému vozidla, energetických systémech, domácích spotřebičích, výtazích), digitální zpracování obrazu a rozpoznávání vzorků [3].

## 3.3 Q-learning

Q-learning je jednoduchý a výkonný algoritmus zpětnovazebního učení. Zpětnovazební učení je takzvané učení ze zkušeností, zpětné vazby [1].

V této metodě učení se agent snaží vytvořit optimální strategii z předchozích zkušeností z prostředí. Přechozí zkušenost je posloupnost odměn(O) za stav(S) a akci(A) –  $S_0, A_0, O_1, S_1, A_1, O_2, S_2, A_2, O_3, S_3, A_3, O_4, S_4 \dots$ . Což znamená, že agent byl ve stavu  $S_0$  a provedl akci  $A_0$ , což vedlo k tomu, že obdržel odměnu  $O_1$  a byl ve stavu  $S_1$ ; poté provedl akci  $A_1$ , obdržel odměnu  $O_2$  a skončil ve stavu  $S_2$ ; poté provedl akci  $A_2$ , obdržel odměnu  $O_3$  a skončil ve stavu  $S_3$ ; a tak dále. Tato historie interakcí je sled zkušeností, kde zkušenost je n-ticí  $\langle S, A, O, S' \rangle$ , což znamená, že agent byl ve stavu  $S$ , činil akci  $A$ , obdržel odměnu  $O$  a šel do stavu  $S'$  (neboli  $S_{t+1}$ ). Tyto zkušenosti jsou využity k tomu, aby se agent naučil, co v daných situacích dělat. Q-learning se zabývá tím, jak by měl agent jednat v prostředí, aby maximalizoval dlouhodobou odměnu [32].

### 3.3.1 Markovův rozhodovací proces

Nedílnou součástí tohoto učení je Markovův rozhodovací proces (MDP) (viz obr. č. 6).

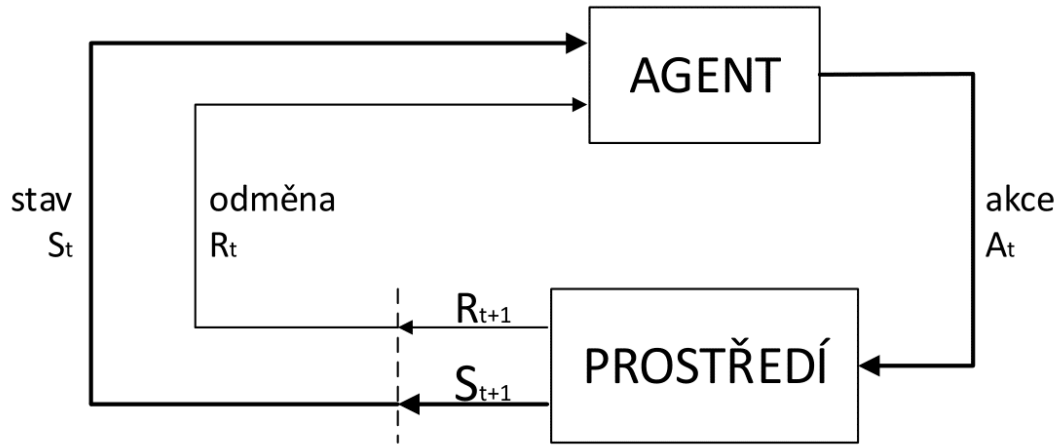
Principem MDP je, že v něm existuje agent, který je zodpovědný za volbu akcí, jelikož interaguje s prostředím, ve kterém se nachází. V každém časovém kroku agent vrátí nějakou reprezentaci stavu systému. S přihlédnutím na tento stav je agentem vybrána další akce. Poté je prostředí přeneseno do dalšího stavu a agentovi je udělena odměna za jeho předchozí akci.

Během tohoto procesu je úkolem agenta maximalizovat získané odměny za zvolené akce. Tímto se vytvoří strategie pro řešení dané problematiky systému. Lze obecně říct, že cílem Q-learningu je naučení agenta optimální strategii. Na začátku učení je akce vybrána náhodně, jelikož Q-tabulka<sup>2</sup> obsahuje jen samé nuly. Po provedení určitého počtu akcí, agent vybírá akce s nejvyšší hodnotou odměny pro maximalizaci odměny [1].

Q-learning nepotřebuje žádný model prostředí a umí najít optimální řešení pro každý konečný Markovův rozhodovací proces [2].

---

<sup>2</sup>jako paměť je použita Q-tabulka, v níž jsou uloženy kvalitativní hodnoty jednotlivých stavů a akcí. Velikost tabulky je dána počtem stavů a možných akcí [1]



Obrázek 6: Markovův rozhodovací proces [34]

### 3.3.2 Řízení Q-learningu

Celý algoritmus je řízen rovnicí

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha * [R_{t+1} + \gamma * \max_{A_{t+1}} Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]. \quad (2)$$

Člen  $Q(S_t, A_t)$  představuje hodnotu ve Q-tabulce pro aktuálně zpracovanou akci  $A_t$  a stav  $S_t$  [1].

Míra učení  $\alpha$  udává do jaké míry budou staré informace přepsány novými. Může nabývat hodnot v intervalu od 0 do 1. Když se  $\alpha = 0$ , tak se agent nebude učit nové informace a bude výhradně používat předchozí poznatky. Naopak, když se  $\alpha = 1$ , tak agent vezme v potaz nově nabitě informace. V praxi má často vysokou hodnotu. Po nalezení optimálního řešení by se měla rovnat 0, což znamená, že by bylo zastaveno učení.

Konstanta  $R_{t+1}$  představuje míru odměny pro zvolenou akci.

Index  $\gamma$ , neboli diskontní faktor, udává jak moc budoucí události ztratí svou hodnotu, podle toho jak daleko v budoucnosti se vyskytují. Nabývá hodnot v intervalu od 0 do 1. Jestliže je  $\gamma = 1$ , odměna obdržená nyní má stejný význam jako odměna obdržená v budoucnosti. Když je  $\gamma = 0$ , agent bude upřednostňovat akce, které povedou k okamžité odměně. To může značně ovlivnit rychlost učení. Hodnota  $\gamma * \max_{A_{t+1}} Q(S_{t+1}, A_{t+1})$  definuje diskontní odhadovanou budoucí odměnu [32].

Zkušenost  $\langle S, A, 0, S' \rangle$  poskytuje datový bod pro hodnoty  $Q(S_t, A_t)$ . Datový bod je skutečná aktuální odměna + diskontní odhadovaná budoucí odměna [32].

Pro strategii učení je důležitá mimo jiné také konstanta náhodného výběru  $\varepsilon$ . Ta zapřičiňuje, že místo výběru akcí na základě maximální odměny je vybrána akce náhodně. Náhodné chování je důležité, protože umožňuje agentovi prozkoumat a objevit nové stavy, které by jinak nebyly

vybrány. Tato konstanta většinou nabývá hodnoty 0,95 (5 % náhodných akcí) nebo 0,98 (2 % náhodných akcí) [33].

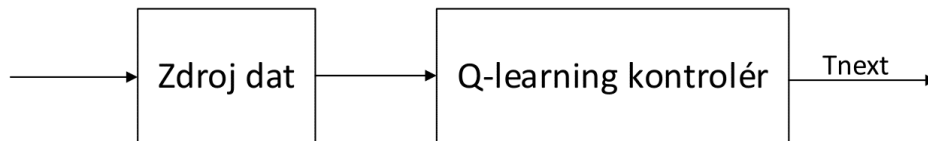
Nastavení konstant závisí na dané strategii, velikosti stavového prostoru a cílovém chování celého systému [1].

### **3.3.3 Využití Q-learningu**

Tento algoritmus nabízí snadnou implementaci. Nabízí dobrou rovnováhu mezi optimalností paměti a energetickými požadavky. Tento algoritmus je vhodné použít pro učení simulací, nebo jinde, kde při vyzvednutí záporné odměny nedojde ke ztrátám [1].

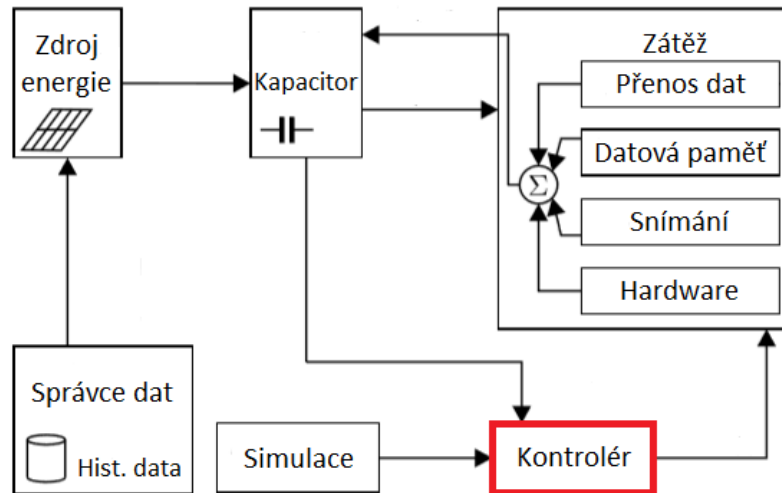
## 4 Návrh simulačního softwaru

Vlastní simulační software se skládá ze dvou částí – zdroje dat a Q-learning kontroléru (viz obr. č. 7). Simulace je spuštěna s prvním záznamem dat  $X_m$  a ukončena s posledním. Mezi tím je činnost simulace řízena časovým úsekem (periodou)  $T_{next}$ . Výstupem simulace jsou data z průběhu simulace uložená do souboru *HWNSF.csv*. Ty obsahují informace o čase ve kterém bylo měřeno, hodnotách vstupních dat, predikovaných hodnotách, periodě měření, stavovém indexu, hodnotě relativní chyby predikce  $\varepsilon_{rel}$  a dané odměně. Data jsou oddělena čárkou a určena pro následné zpracování.



Obrázek 7: Blokové schéma návrhu simulačního softwaru

Vlastní simulační software navazuje na již vytvořený simulační framework (viz obr. č. 8), ve kterém byl vytvořen zcela nový modul kontroléru (červeně označeno).



Obrázek 8: Blokové schéma simulačního frameworku [29]

## 4.1 Zdrojová data pro simulaci

Data  $X_m$  ze kterých systém vychází jsou meteorologická data obsahující informaci o slunečním svitu (viz tabulka č. 2). Byla sbírána po dobu 5 let v období let 2008–2012 při kontinuální monitoringu s periodou pět minut. Tato data jsou výstupním parametrem *Zdroje dat* a jedním z vstupních parametrů  $Q$  – *learning kontroléru*. Dalším výstupním parametrem je konstanta stupně polynomu  $N$  (viz podkapitola č. 4.2.2).

Tabulka 2: Ukázka zdrojových dat

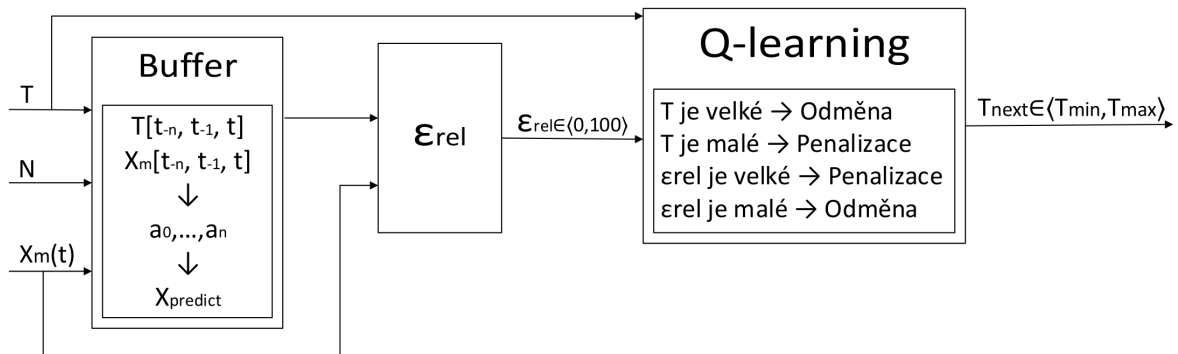
Rok	Den	Čas	Energie [J]
2008	1	0:00	9,310
2008	1	0:05	5,338
2008	1	0:10	5,240
2008	1	0:15	4,236

Do bloku *Zdroj dat* vstupuje časový úsek (perioda)  $T$ , která je výstupem  $Q$  – *learning kontroléru* z předchozí iterace ( $T_{next}$ ).

## 4.2 Návrh Q-learning kontroléru

Modul  $Q$  – *learning kontroléru* obsahuje algoritmus, jehož cílem je umět předvídat následný vývoj, v této práci konkrétně atmosferických podmínek prostředí, ve kterém se nachází a dle toho ovládal systém. Vstupem kontroléru jsou zdrojová data (viz. podkapitola č. 4.1). Výstupem je časový úsek  $T_{next}$ , který řídí čas provedení další iterace. Návrh  $Q$  – *learning kontroléru* je blíže specifikován v kapitole č. 4.2.

$Q$  – *learning kontrolér* implementuje strategii řízení simulace. Skládá se z několika bloků (viz obr. č. 9) – *Buffer*,  $\epsilon_{rel}$  a  $Q$  – *learning* se sadou pravidel. Vstupy jsou data s informací o sluneční energii  $X_m$ , času měření  $T$  a stupni polynomu  $N$ . Výstupem modulu je časový úsek  $T_{next}$ , jenž určuje čas provedení další iterace.

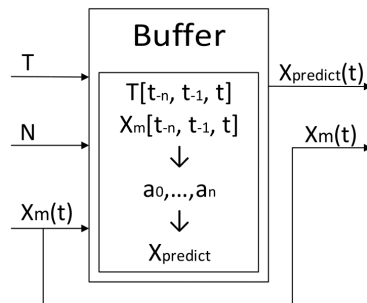


Obrázek 9: Blokové schéma návrhu Q-learning kontroléru



### 4.2.1 Blok Buffer

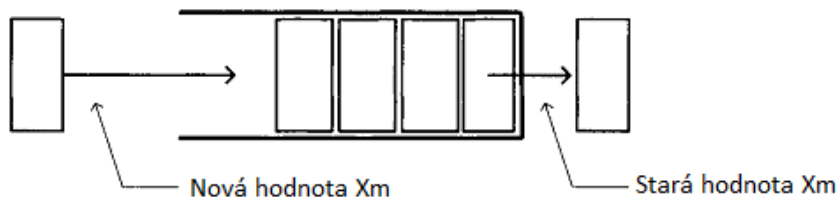
Blok *Buffer* (viz obr. č. 10) má dvě funkce. První z nich je, že funguje jako uložště. Druhou funkcí je počítání predikované hodnoty  $X_{predict}$ .



Obrázek 10: Blok Buffer

Blok *Buffer* má v sobě uloženo několik předchozích hodnot  $X_m$ , což jsou aktuálně naměřené hodnoty snímače, jejichž množství závisí na řádu polynomu  $N$  (viz podkapitola č. 4.2.2). Je zde použita metoda tzv. *FiFoBuffer* (viz obr. č. 11). Tento buffer ukládá data, dokud nedocílí počtu  $N + 1$  dat. Poté nejstarší z dat ze začátku vymaže, ostatní data se posunou o 1 místo vpravo a poslední prázdné místo vlevo na konci je zaplněno nejnovější hodnotou  $X_m$ .

Metodou *FiFoBuffer* je taktěž ukládána hodnota  $T$ , což je čas ve kterém se hodnota  $X_m$  změnila.



Obrázek 11: Princip FiFoBufferu [28]

Druhou funkcí bloku *Buffer* je počítání predikované hodnoty  $X_{predict}$ . Výpočet je založen na aproximaci polynomem daného stupně  $N$  (viz podkapitola č. 4.2.2). Stupeň polynomu bude zvolen experimentálně.

### 4.2.2 Aproximace polynomem pro získání predikované hodnoty $X_{predict}$

Aproximace polynomem je použita pro výpočet predikované hodnoty  $X_{predict}$ .

Pod pojmem polynom rozumíme funkci ve tvaru:

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0, \quad (3)$$

kde  $a_0, a_1, \dots, a_n$  jsou reálná čísla, tzv. koeficienty polynomu. Je-li navíc  $a_n \neq 0$ , říkáme, že  $n$  je stupeň polynomu  $p$ .

$x^n$  jsou hodnoty proměnné.

Nejvyšší mocnina, která se v polynomu objevuje definuje stupeň polynomu  $n$  (v této práci dále značen velkým písmenem  $N$ ) [25].

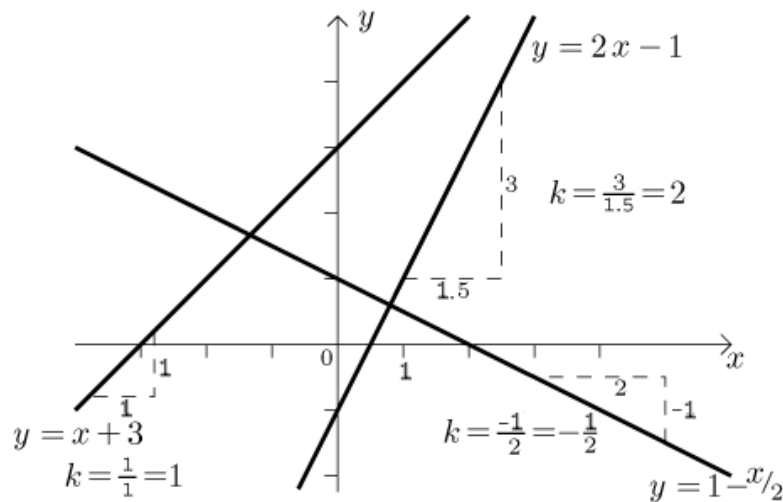
Aproximace je nepřesné vyjádření nějaké funkce. Jedná se o alternativu analytického řešení, kdy není dostatek informací nebo je to příliš náročné na provedení [26]. Aproximace polynomem představuje proložení daných hodnot polynomem. Pro získání koeficientů polynomu bylo využito metody nejmenších čtverců. Tyto hodnoty jsou vypočteny tak, aby součet druhých mocnin odchylek původních hodnot od získaného polynomu byl minimální (viz rovnice č. 4) [27].

$$F = \sum_{i=1}^n e_i^2 \rightarrow \min, \quad (4)$$

kde  $e_i$  je odchylka původní hodnoty od získaného polynomu (viz rovnice č. 5).

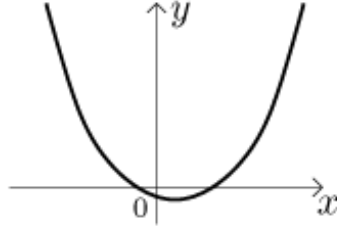
$$e_i = x_i - p(x_i). \quad (5)$$

- Aproximace lineárním polynomem – Jedná se o proložení daných hodnot lineární funkcí ve tvaru  $f(x) = a_1 x + a_0$ , kde  $a_i$  jsou koeficienty polynomu a  $x$  je hodnota proměnné. Grafem lineárního polynomu je přímka (viz graf na obr. č. 12) [25].



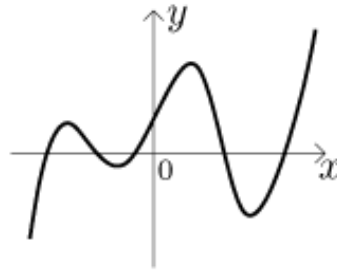
Obrázek 12: Graf lineárního polynom [25]

- Polynom 2. stupně – Jedná se o proložení bodů kvadratickou funkcí (parabolu – viz graf na obr. č. 13) s rovnicí  $f(x) = a_2x^2 + a_1x + a_0$ , kde  $a_i$  jsou koeficienty polynomu a  $x_i$  je hodnota proměnné.



Obrázek 13: Graf polynomu 2. stupně [25]

- Polynom 5. stupně – Funkce polynomu 5. stupně je ve tvaru  $f(x) = a_5x^5 + a_4x^4 + \dots + a_0$ , kde  $a_i$  jsou koeficienty polynomu a  $x_i$  je hodnota proměnné. Grafem polynomu je na obr. č. 14.



Obrázek 14: Graf polynomu 5. stupně [25]

Z uložených dat  $X_m$  a času  $T$ , ve kterém byla hodnota  $X_m$  změřena, je podle rovnice (záleží na volbě stupně polynomu) dané funkce vytvořena soustava rovnic. Tato soustava je poté převedena do maticového zápisu. Z matice je po úpravách získáno řešení – koeficienty polynomu, které dají za vznik výslednému tvaru polynomu daného stupně. Do takto vytvořeného polynomu jsou dosazeny hodnoty  $X_m$  a  $T$ , a je vypočítána výsledná hodnota  $X_{predict}$ .

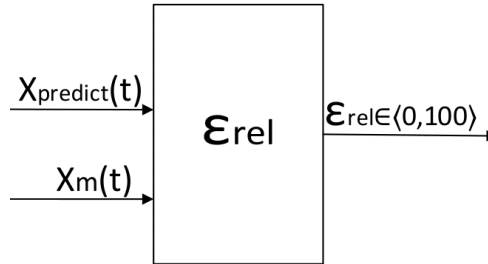
#### 4.2.3 Blok $\varepsilon_{rel}$

Následuje blok, který nese název  $\varepsilon_{rel}$  (viz obr. č. 15) a jenž vypočítává relativní chybu predikce podle rovnice

$$\varepsilon_{rel} = \frac{|X_m - X_{predict}|}{X_{max}} * 100, \quad (6)$$

kde se v čitateli vypočte absolutní rozdíl naměřené hodnoty  $X_m$  a predikované hodnoty  $X_{predict}$ . Tato hodnota je vydělena celým rozsahem  $X_{max}$  daného senzoru. Výsledkem je normalizovaný

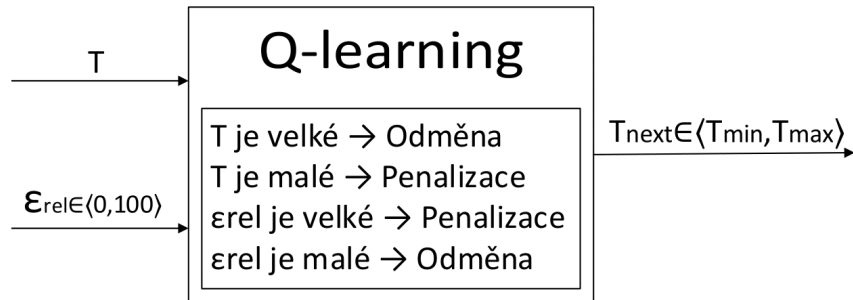
rozsah, který nabývá hodnot  $\langle 0; 100 \rangle$  %. Hodnota 100 % znamená, že se predikovaná hodnota od skutečně naměřené liší o celý rozsah. Hodnota 0 % znamená, že se neliší vůbec.



Obrázek 15: Blok  $\varepsilon_{rel}$

#### 4.2.4 Blok Q-learning

Dalším blokem je algoritmus *Q-learning* se sadou pravidel (viz obr. č. 16). Jeho úkolem je naučit se predikovat čas  $T_{next}$  dalšího měření senzoru, aniž by došlo ke ztrátě důležitých dat a zároveň, aby nedocházelo k jejich redundanci.



Obrázek 16: Blok Q-learning se sadou pravidel

Jelikož se jedná o metodu zpětnovazebního učení,  $T_{next}$  je vyhodnoceno až projde simulací. Při započatí simulace jsou první hodnoty  $X_{predict}$  náhodné, jelikož algoritmus nemá žádnou předchozí zpětnou vazbu, ze které by mohl vycházet. Postupem času ji však dostává, čímž se přesnost predikce časového úseku zvyšuje.

*Q-learning* má dvě vstupní proměnné, stavy a akce. Mezi stavy se řadí hodnoty  $\varepsilon_{rel}$ , viz tabulka 4, ve které jsou jednotlivým hodnotám přiřazeny stavové odměny. Akce jsou jednotlivé hodnoty  $T$ , viz tabulka 3, ve které jsou taktéž jako u  $\varepsilon_{rel}$ , jednotlivým hodnotám přiřazeny akční odměny.

Cílem je, aby senzor měřil, co nejméně. Z toho vychází následující sada pravidel:

- Když bude  $T$  velké, dostane agent odměnu, jelikož cílem je, aby měřil v co největších intervalech.

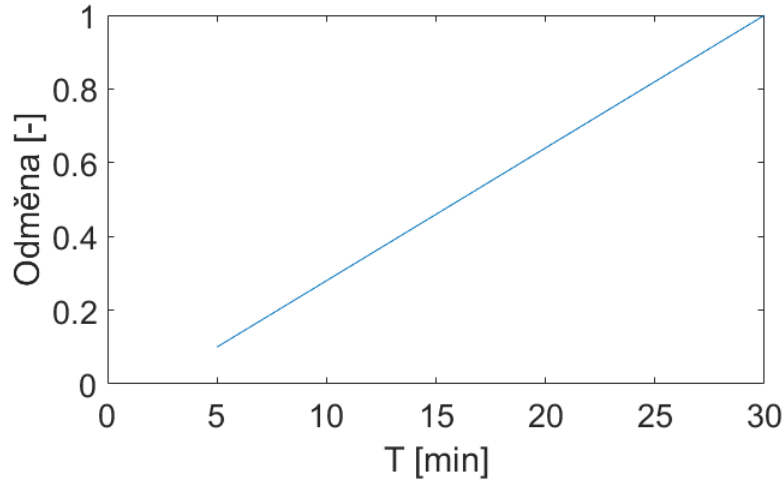
- Když bude  $T$  malé, dostane agent penalizaci, protože se chceme vyvarovat malému vzorkování.
- Když bude  $\varepsilon_{rel}$  velké, tak se predikovaná hodnota výrazně liší od reálně naměřené hodnoty, tzn. že predikce byla neúspěšná – dostane penalizaci.
- Když bude  $\varepsilon_{rel}$  malé, tak se predikovaná hodnota tolik neliší od reálně naměřené, tzn. že predikce byla úspěšná – dostane odměnu.

#### 4.2.5 Návrh kombinací expertních konstant a ostatních parametrů Q-learningu

Z výše popsaného vychází, že výstupem Q-learning systému je časový úsek  $T_{next}$ , což je doba za kterou proběhne další iterace simulace s periodou  $T$  (viz rovnice 7), kde  $n$  je krok.  $T_{next}$  nabývá hodnot  $T_{next} \in \langle T_{min}; T_{max} \rangle (min)$ . Toto tvrzení je platné pouze, když polynom je validní<sup>3</sup>, jinak  $T_{next} = T_{min}$ .

$$T_{next} = T(n + 1) \quad (7)$$

Rozsah hodnot  $T$  je od 5 do 30 minut, viz graf na obr. č. 17.



Obrázek 17: Graf závislosti časového úseku  $T$  na odměně

Jelikož cílem algoritmu je použití, co nejvyšší hodnoty  $T$ , aby systém měřil nejlépe, co 30 min. *Odměna* je u 30 min nastavena na nejvyšší hodnotu, 1. Čím se hodnota  $T$  snižuje, snižuje se taky lineárně *Odměna*, viz 17. Hodnoty jednotlivých  $T$  a jejich odměn jsou uvedeny v tabulce 3. Hodnoty odměn byly vypočítány na základě rovnice pro přímku, která vypadá následovně:

$$Odměna_{Akce} = 0.036 * T - 0.08 \quad (8)$$

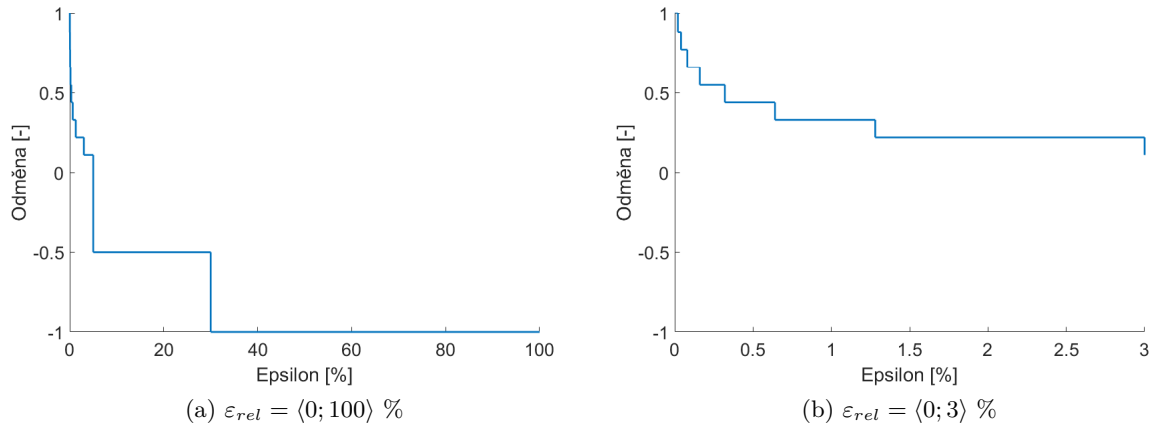
$\varepsilon_{rel}$  je normalizovaný rozsah, který nabývá hodnot  $\langle 0; 100 \rangle \%$ . Narozdíl od hodnoty  $T$ , která dostávala odměnu za své nejvyšší hodnoty,  $\varepsilon_{rel}$  dostává odměnu (tzv *OdměnaStav*) za nejnižší

<sup>3</sup>musí splňovat podmínku, že hodnot  $Xm$  je v bufferu uloženo  $N + 1$  ( $N$  je stupeň polynomu)

Tabulka 3: Hodnoty jednotlivých akcí (periody  $T$ ) a jejich odměny

T [min]	OdměnaAkce [-]
5	0.10
10	0.28
15	0.46
20	0.64
25	0.82
30	1.00

hodnoty (viz tabulka 4). Při hodnotách od 0 – 0.02 % je odměna systému největší, jelikož  $\varepsilon_{rel}$  signalizuje, že se predikovaná hodnota tolik neliší od reálně naměřené, tzn. že predikce byla úspěšná. Čím se hodnota  $\varepsilon_{rel}$  snižuje, snižuje se také odměna, viz 18. Hodnoty  $\varepsilon_{rel}$  od 5 – 100 % jsou vnímány negativně, jelikož dokazují fakt, že se predikovaná hodnota liší od reálně naměřené hodnoty, za což dostává penalizaci v podobě záporné odměny. Všechny intervaly  $\varepsilon_{rel}$  a jejich odměny/penalizace jsou uvedeny v tabulce 4.



Obrázek 18: Graf závislosti relativní chyby predikce  $\varepsilon_{rel}$  na odměně

Vynásobení dílčích odměn (stavových a akčních) dá výslednou odměnu systému za vykonanou predikci (viz rovnice č. 9).

$$Odměna = OdměnaStav * OdměnaAkce \quad (9)$$

Pro testování simulačního softwaru bude použito několik kombinací expertních konstant a stupňů polynomu (viz tabulka č. 5).

Rychlost učení  $\alpha$  byla zvolena experimentálně: 0,2 a 0,5.

Diskontní faktor  $\gamma$  byl nastaven na 0,1 a konstanta výběrů náhodných akcí  $\varepsilon$  0,95. Tyto konstanty se po čas testování neměnily.

Dalším parametrem je stupeň polynomu  $N$ , který byl taktéž volen experimentálně: 1, 2 a 5.

Tabulka 4: Hodnoty jednotlivých stavů (relativní chyba predikce  $\varepsilon_{rel}$ ) a jejich odměny

$\varepsilon_{rel}$ [%]	OdměnaStav[-]
$\langle 0; 0.02 \rangle$	1.00
$\langle 0.02; 0.04 \rangle$	0.88
$\langle 0.04; 0.08 \rangle$	0.77
$\langle 0.08; 0.16 \rangle$	0.66
$\langle 0.16; 0.32 \rangle$	0.55
$\langle 0.32; 0.64 \rangle$	0.44
$\langle 0.64; 1.28 \rangle$	0.33
$\langle 1.28; 3 \rangle$	0.22
$\langle 3; 5 \rangle$	0.11
$\langle 5; 30 \rangle$	- 0.50
$\langle 30; 100 \rangle$	- 1.00

Rozsah senzoru byl určen z analýzy přejatých dat, kde byla maximální hodnota rovna 1298 J.

### 4.3 Implementace Q-learning kontroléru

Implementace probíhávala v programovacím jazyku *C#.NETCore*.

Byla vytvořena třída *QLearningControllerPolyfit*, ve které byly deklarovány proměnné expertních a jiných konstant Q-learningu a také naprogramován algoritmus, který řídí činnost kontroléru. Druhou vytvořenou třídou byla třída *Polyfit*, která vrací koeficienty polynomu. Poslední byla třída *FiFoBuffer*, která slouží jako uložisko dat pro aproximaci polynomem.

#### 4.3.1 Deklarace proměnných

Byly deklarovány proměnné pro expertní konstanty Q-learningu, které tvoří strategii jeho učení:

- rychlost učení  $\alpha$ ,
- diskontní faktor  $\gamma$ ,
- výběr náhodných akcí  $\varepsilon$ .

Dalšími parametry, pro které bylo nutno deklarovat, byly:

- stupeň polynomu  $N$ ,
- akce a jejich odměny,
- stavy a jejich odměny,
- rozsah senzoru.

#### 4.3.2 Třída *QLearningControllerPolyfit*

Třída *QLearningControllerPolyfit* zahrnuje všechny dílčí části, jenž vytváří kontrolér (viz kapitola 4.2) – Buffer,  $\varepsilon_{rel}$  a Q-learning se sadou pravidel. Obsahuje metody:

- *GetIntervalIndex* – metoda, která slouží pro záznam indexu stavu  $\varepsilon_{rel}$ , který byl použit.
- *EpsilonErr* – metoda, ve které dochází k výpočtu hodnoty  $\varepsilon_{rel}$ . Výpočet je poté využit v následující metodě, která se nazývá *EnviromentFeedback*.
- *CalcPolynomValue* – vypočítá výslednou hodnotu polynomu,  $X_{predict}$  (viz podkapitola č. 4.3.3).
- *EnviromentFeedback* – Tato metoda má mnohou funkcí. Jednou z nich je volání třídy *Polyfit*, která obsahuje metodu *DoComputeCoefficients*, jenž vypočítá jednotlivé koeficienty polynomu, které jsou poté zpracovány pomocí metody *CalcPolynomValue* v hodnotu  $X_{predict}$ . Následující funkcí je použití metody *EpsilonErr*, jejíž funkce je popsána výše. Další funkcí je volání metody *GetIntervalIndex*, která udává index stavu  $\varepsilon_{rel}$ , který byl použit. V neposlední řadě je zde vypočítávána výsledná odměna za zvolenou akci a stav.

Implementace samotného algoritmu Q-learningu byla převzata od [29]. Byly deklarovány pouze stavy, akce a expertní konstanty, které jsou jeho součástí.

#### 4.3.3 Třída *Polyfit*

Třída *Polyfit* vrací koeficienty pro aproximaci polynomem  $p(x)$  stupně  $N$ . Koeficienty polynomu jsou v sestupném pořadí, jejich délka je  $N + 1$  a jsou dále zpracovány v metodě *CalcPolynomValue*. Tato metoda slouží pro výpočet výsledné hodnoty polynomu, což je hodnota predikce  $X_{predict}$ .

#### 4.3.4 Třída *FiFoBuffer*

Třída *FiFoBuffer* byla vytvořena pro ukládání dat  $X_m$  a  $T$ . A to tím způsobem, že je ukládán  $N + 1$  počet hodnot. Poté se nejstarší z dat ze začátku vymaže, ostatní data se posunou o 1 místo vpravo a poslední prázdné místo vlevo na konci je zaplněno nejnovější hodnotou. Byl vytvořen zvlášť *FiFoBuffer* pro hodnoty času  $T$  a zvlášť pro hodnoty  $X_m$ .



## 5 Metrika vyhodnocení

Pro statistické porovnání dat ze simulace s výchozími daty budou použity dvě metody. První bude k porovnávání dvou výběrových souborů využívat parové testy. Druhou metodou je výpočet střední kvadratické chyby, MSE a odmocniny ze střední kvadratické chyby, RMSE.

Pro vyhodnocení výsledků testování akcí, stavů, odměn a přesnosti predikce bude využita kvartilová analýza a analýza četností.

Pro statistické vyhodnocení bude využit software *RStudio*.

### 5.1 Párové testy

Pomocí párových testů je možno rozhodnout, zda jsou data dvou souborů shodná či nejsou. Funguje to tím způsobem, že je jeden soubor měřený za dvou různých podmínek. Cílem je rozhodnout, zda tyto podmínky ovlivňují měřenou hodnotu. Bude porovnáván soubor s odchylkami obou výběrů.

Při výběru testu, pro porovnávání dvou výběrových souborů pomocí jejich odchylek, hraje důležitou roli skutečnost, zda odchylky daných datasetů pocházejí z normálního rozdělení, či nikoliv. K ověření této skutečnosti bude použit Shapirův–Wilkův test normality (S–W test). S–W test testuje hypotézu, zda náhodný výběr  $X_1, \dots, X_n$  pochází z normálního rozdělení s parametry  $N(\mu, \sigma^2)$ , a to na základě skutečnosti, zda body Q–Q grafu<sup>4</sup> se významně odlišují od regresivní přímky proložené těmito body. Leží-li body přibližně na přímce, jedná se o normální rozdělení. Pokud neleží, nejedná se o normální rozdělení. Kromě tohoto empirického posouzení, lze využít přednastavené funkce Shapiro–Wilkova testu v softwaru RStudio, která umí exaktně ověřit normalitu.

V případě, že rozdíl (odchylka) výběrů nesplňuje podmínku normality, přistupuje se k použití mediánového testu, tzv. Wilcoxonova testu. U něhož testujeme hypotézu  $H_0$ , která udává, že rozdíly (odchylky  $X_m$  a  $X_{predict}$ ) jsou symetricky rozloženy kolem 0, tzn., že součet kladných a záporných odchylek by měl být roven 0 a tudíž, že se datové soubory od sebe statisticky významně neliší. Alternativní hypotéza  $H_A$  je poté negací nulové, tzn., že rozdíly (odchylky  $X_m$  a  $X_{predict}$ ) nejsou symetricky rozloženy kolem 0 a datové soubory se od sebe statisticky významně liší.

Pokud budou výběry splňovat podmínku normality, znamená to, že je třeba přistoupit k použití tzv. párového t-testu. U něhož testování probíhá obdobně jako u Wilcoxonova testu.

Výsledkem testů je tzv. p-hodnota<sup>5</sup>. Bude-li výsledkem testů p-hodnota větší než 5 %, potvrzujeme  $H_0$ , což znamená, že se data statisticky významně neliší. Pokud bude p-hodnota menší než 5 %<sup>6</sup>, zamítá se nulová hypotéza ve prospěch alternativní a znamená to, že data se statisticky významně liší [23].

<sup>4</sup>grafické posouzení toho, zda data pocházejí z nějakého známého rozdělení

<sup>5</sup>p-hodnota je nejvyšší hladina významnosti na které nezamítám nulovou hypotézu

<sup>6</sup>nejnižší možná hladina na které je ještě možné zamítnout nulovou hypotézu

Postup vyhodnocení pomocí párového testování:

1. Je navržena nulová  $H_0$  a alternativní hypotéza  $H_A$ .
2. Jsou odstraněny páry dat  $X_m$  a  $X_{predict}$ , kde  $X_{predict}$  neměřilo.
3. Jsou vypočítány odchylky mezi páry  $X_m$  a  $X_{predict}$  v daném čase.
4. Je vybráno 5000 náhodných datových párů.
5. Je ověřena normalita odchylek.
6. Podle výsledné normality je použit t-test nebo Wilcoxonův test.
7. Je vyhodnocen výsledek testu podle p-hodnoty, mediánu a intervalového odhadu.
8. Je potvrzena nebo zavrhnuta nulová hypotéza.

## 5.2 MSE, RMSE a SD

MSE (Mean Squared Error) neboli střední kvadratická chyba je lineární derivace, která porovnává plochu pod křivkami, čímž udává rozdílnost mezi nimi. Vypočítá se následnou rovnicí:

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (10)$$

kde  $n$  je celkový počet odchylek,

$Y_i$  a  $\hat{Y}_i$  jsou jednotlivá data, které jsou porovnávána.

Díky výsledku této rovnice zjistíme podobnost dvou výběrů. Čím je hodnota MSE menší, tím jsou si výběry ( $Y_i$  a  $\hat{Y}_i$ ) více podobné. V ideálním případě by byla hodnota rovna 0, což by znamenalo, že jsou výběry totožné.

RMSE (Root Mean Squared Error) je odmocnina ze střední kvadratické chyby (viz rovnice 11). Jedná se o nejpřirozenější aproximaci standartní chyby z konečného počtu realizací, která je úzce propojena se standartní odchylkou chyby odhadu. Taktéž jako MSE udává podobnost mezi jednotlivými výběry ( $Y_i$  a  $\hat{Y}_i$ ) a také je ideálně rovna 0 (absolutní shoda) [22].

$$RMSE = \sqrt{MSE} \quad (11)$$

SD (standard deviation) neboli směrodatná odchylka je ukazatel odchýlení/rozptýlení hodnot od průměru hodnot. Je-li SD malá, jsou si prvky souboru většinou navzájem podobné, a naopak velká značí velké vzájemné odlišnosti. Výpočet probíhá podle rovnice 12, kde  $x_i$  a  $\hat{x}_i$  jsou porovnávána data [24].

$$SD = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2} \quad (12)$$

V této práci budou porovnávány rozdíly mezi naměřenými hodnotami simulací a hodnotami původními. Vytvoří se páry těchto hodnot, jedna hodnota  $X_{predict}$  a jedna hodnota  $X_m$  ve stejném časovém okamžiku. Mezi těmito páry budou vypočteny odchylky a dále budou porovnávány výše uvedenými metodami.

### 5.3 Kvartilová analýza

Kvartil je typem kvantilu. Jedná se o rozdělení datového souboru do čtyř stejných částí, z nichž každá představuje čtvrtinu vzorku dat. Existují tři kvartily a to:

- 1.  $Q_v$  (dolní kvartil) – oddělí nejmenších 25 % dat od nejvyšších 75 %.
- 2.  $Q_v$  (střední kvartil neboli medián) – rozdělí data na dvě poloviny.
- 3.  $Q_v$  (horní kvartil) – oddělí nejmenších 75 % dat od nejvyšších 25 % [30].

V práci budou díky této analýze vyhodnoceny výsledky testování akcí, stavů, odměn a přesnosti predikce.

### 5.4 Analýza četnosti

Četnost je statistický parametr, který udává, kolik hodnot daného statistického znaku se vyskytuje ve statistickém souboru. Udává se absolutně nebo relativně (vzhledem k celkovému počtu prvků) [31]. Touto analýzou budou v práci vyhodnoceny výsledky testování periody  $T$ .

## 6 Testování a vyhodnocení výsledků

Pro testování správné funkce simulace bylo vytvořeno šest kombinací expertních konstant a stupňů polynomu (viz tabulka č. 5), které byly definovány v podkapitole č. 4.2.5.

Tabulka 5: Kombinace jednotlivých expertních konstant a stupně polynomu

$\alpha$	$N$	$\gamma$	$\varepsilon$
0,2	1	0,1	0,95
0,2	2	0,1	0,95
0,2	5	0,1	0,95
0,5	1	0,1	0,95
0,5	2	0,1	0,95
0,5	5	0,1	0,95

### 6.1 Testování akcí Q-learningu – Perioda $T$

Perioda  $T$  nabývá hodnot od 5 do 30 min a udává četnost měření v simulaci. Z tabulky č. 6 vychází, že stupně polynomu  $N$  výrazně ovlivňuje celkový počet měření a to takovým způsobem, že s zvětšujícím se  $N$  se zvyšuje celkový počet naměřených dat. V porovnání  $\alpha = 0,2$   $N = 1$  s  $\alpha = 0,2$   $N = 5$  až o 17,93 % (viz obr. č. 19). Samotná rychlost učení  $\alpha$  jej ovlivňuje také, ale zanedbatelně. V porovnání  $\alpha = 0,2$   $N = 1$  se celkový počet naměřených dat u  $\alpha = 0,5$   $N = 1$  zvýšil o 0,10 %.

Tabulka 6: Celkový počet měření při jednotlivých kombinacích konstant

$\alpha$	$N$	Celkový počet měření
0,2	1	102593
0,2	2	107804
0,2	5	125002
0,5	1	102702
0,5	2	107977
0,5	5	124934

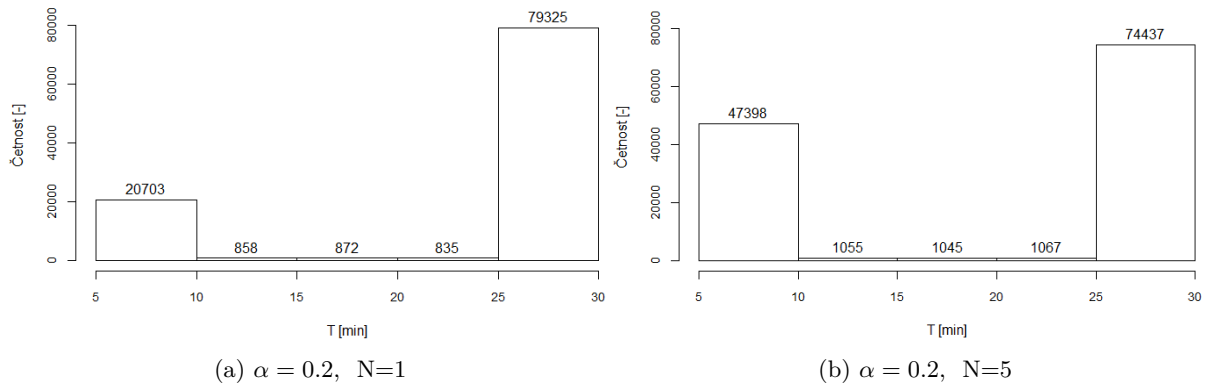
Tabulka 7: Absolutní (relativní) četnost period  $T$  při jednotlivých kombinacích konstant

$\alpha$	$N$	$T[\text{min}]$		
		5	10	15
0,2	1	19848 (19,35 %)	855 (0,83 %)	858 (0,84 %)
0,2	2	25838 (23,97 %)	914 (0,85 %)	925 (0,86 %)
0,2	5	46350 (37,08 %)	1048 (0,84 %)	1055 (0,84 %)
0,5	1	19922 (19,40 %)	862 (0,84 %)	900 (0,88 %)
0,5	2	26228 (24,29 %)	878 (0,81 %)	892 (0,83 %)
0,5	5	46245 (37,02 %)	1064 (0,85 %)	1065 (0,85 %)

Tabulka 8: Absolutní (relativní) četnost period  $T$  při jednotlivých kombinacích konstant

$\alpha$	$N$	$T[\text{min}]$		
		20	25	30
0,2	1	872 (0,85 %)	835 (0,81 %)	79325 (77,32 %)
0,2	2	1260 (1,17 %)	938 (0,87 %)	77929 (72,29 %)
0,2	5	1045 (0,84 %)	1067 (0,85 %)	74437 (59,55 %)
0,5	1	929 (0,90 %)	850 (0,83 %)	79239 (77,15 %)
0,5	2	934 (0,86 %)	921 (0,85 %)	78124 (72,35 %)
0,5	5	1051 (0,84 %)	1081 (0,87 %)	74428 (59,57 %)

Největší četnost zastupuje perioda 30 min (např. 79325 (77,32 %) u  $\alpha = 0,2$   $N = 1$ ), což je vnímáno velmi pozitivně, jelikož cílem algoritmu je, aby měřil, co nejméně, tzn. s co největší možnou periodou. Druhou nejvyšší hodnotou je perioda 5 min (např. 19848 (19,35 %) u  $\alpha = 0,2$   $N = 1$ ). Poté následují ostatní periody od 10 do 25, u nichž se hodnota četnosti statisticky významně neliší.

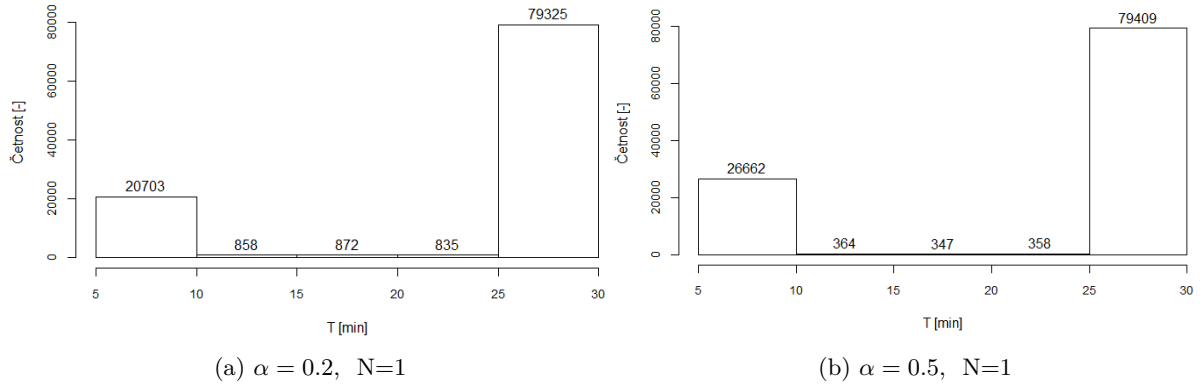


Obrázek 19: Histogram periody  $T$  v dané kombinaci konstant

S přibývajícím stupněm polynomu  $N$  se taktéž výrazně zvyšuje četnost samotné periody  $T = 5\text{min}$ . V případě porovnání  $\alpha = 0,2$   $N = 1$  s  $\alpha = 0,2$   $N = 5$  až o 57,18 %. Četnost ostatních period se v rámci změny kombinací konstant statisticky výrazně nemění.

Při porovnání kombinací stejné hodnoty stupně polynomu  $N$  a změně  $\alpha$ , např.  $\alpha = 0,2$   $N = 1$  a  $\alpha = 0,5$   $N = 1$  lze na grafech obr. č. 20 pozorovat, že se četnost jednotlivých period  $T$  výrazně nezměnila.

Výše uvedené poznatky potvrzuje tabulka č. 9, kde průměrná hodnota (MEAN) periody  $T$  se se zvyšujícím stupněm polynomu  $N$  snižuje, což zapříčiňuje vyšší četnost 5 min periody. Tento fakt potvrzuje hodnota prvního kvartilu u  $\alpha = 0,2$   $N = 5$  (1.Qv=5). Oproti tomu je hodnota prvního kvartilu u  $\alpha = 0,2$   $N = 1$  (1.Qv=30).



Obrázek 20: Histogram periody  $T$  v dané kombinaci konstant

Tabulka 9: Hodnoty kvartilů periody  $T$  při jednotlivých kombinacích konstant

$\alpha$	$N$	$T[\text{min}]$					
		MIN	1.Qv	2.Qv	MEAN	3.Qv	MAX
0,2	1	5	30	30	24,75	30	30
0,2	2	5	15	30	23,55	30	30
0,2	5	5	5	30	20,31	30	30
0,5	1	5	30	30	24,72	30	30
0,5	2	5	10	30	23,51	30	30
0,5	5	5	5	30	20,32	30	30

## 6.2 Testování stavů Q-learningu – Relativní chyba predikce $\varepsilon_{rel}$

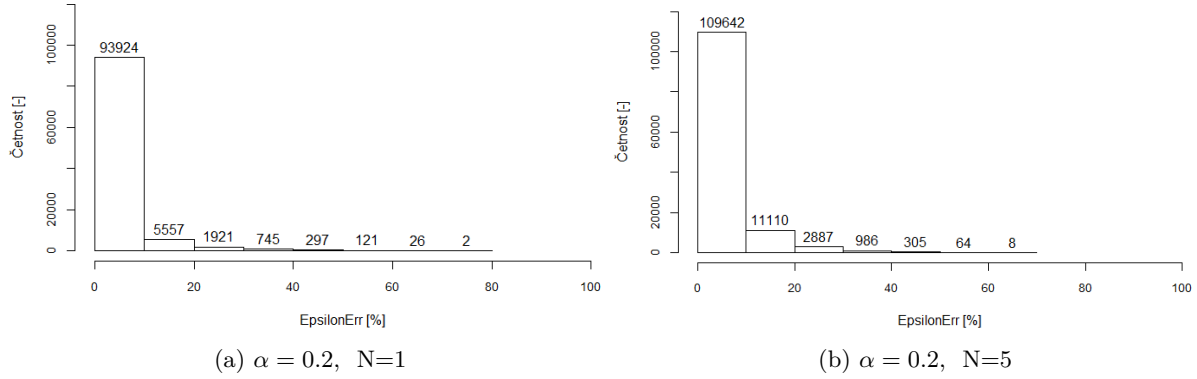
Relativní chyba predikce  $\varepsilon_{rel}$  nabývá hodnot od 0 do 100 %. Hodnota 0 % signalizuje absolutní shodu mezi hodnotami  $X_m$  a  $X_{predict}$ . Hodnota 100 % oproti tomu signalizuje, že se odhad  $X_{predict}$  liší o celý rozsah.

Tabulka 10: Hodnoty kvartilů relativní chyby predikce  $\varepsilon_{rel}$  při jednotlivých kombinacích konstant

$\alpha$	$N$	$\varepsilon_{rel}[\%]$					
		MIN	1.Qv	2.Qv	MEAN	3.Qv	MAX
0,2	1	0	0	0,8475	3,2367	3,8983	72,7350
0,2	2	0	0	1,6720	3,7870	4,8960	71,1830
0,2	5	0	0,0342	3,7227	4,9053	6,6764	69,4157
0,5	1	0	0	0,8475	3,2182	3,8829	72,7350
0,5	2	0	0	1,6950	3,8120	4,9380	70,4390
0,5	5	0	0,0330	3,7310	4,9180	6,6890	65,0460

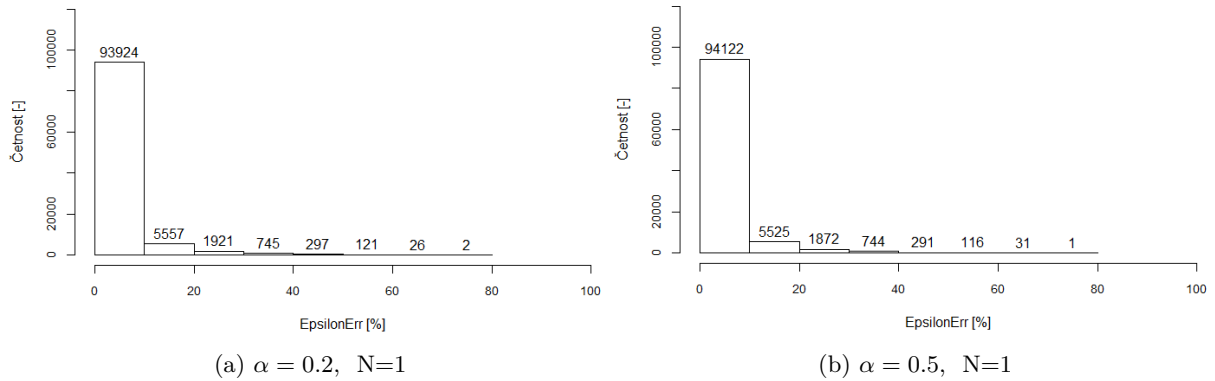
Vliv stupně polynomu  $N$  na relativní chybu predikce  $\varepsilon_{rel}$  je takový, že se zvyšujícím se  $N$  se zvyšuje  $\varepsilon_{rel}$ . A to je viditelné na příkladu porovnání průměrné hodnoty (MEAN)  $\varepsilon_{rel}$  u  $\alpha = 0,2$   $N = 1$  s  $\alpha = 0,2$   $N = 5$  až o 34,02 %. Maximální hodnoty (MAX)  $\varepsilon_{rel}$  se paradoxně zvyšují se snižujícím se  $N$ . Maximální hodnota u  $\alpha = 0,5$   $N = 1$  je v porovnání s  $\alpha = 0,5$   $N = 5$  o 11,82

% větší. U  $\alpha = 0,2$   $N = 1$  a  $\alpha = 0,2$   $N = 5$  pouze o 4,78 %. To značí, že rozsah nepřesnosti predikce je u kombinací s nižšími stupni polynomu větší než u těch, jenž mají stupeň polynomu vyšší. Významně se liší, ale pouze u vyšší rychlosti učení  $\alpha = 0,5$ .



Obrázek 21: Histogram relativní chyby predikce  $\varepsilon_{rel}$  v dané kombinaci konstant

Na grafech obr. č. 21 je vidět rozdíl v hodnotách četnosti v rozmezí od 0 do 20 %. Přisuzuji to faktu, že při kombinacích  $\alpha = 0,2$   $N = 5$  simulace naměřila o 22409 hodnot více než u kombinace  $\alpha = 0,2$   $N = 1$ . Dalším faktem je to, že maximální hodnota kombinace  $\alpha = 0,2$   $N = 1$  je o 10 % vyšší než u  $\alpha = 0,2$   $N = 5$ . To by mohlo značit, že je  $\alpha = 0,2$   $N = 1$  méně přesnou simulací. Není tomu tak, jelikož první a druhý kvartil má menší hodnoty  $\varepsilon_{rel}$  než u  $\alpha = 0,2$   $N = 5$  (viz tabulka č. 10).



Obrázek 22: Histogram relativní chyby predikce  $\varepsilon_{rel}$  v dané kombinaci konstant

Grafy na obr. č. 22 potvrzují fakt, že výraznější vliv na hodnotu  $\varepsilon_{rel}$  má stupeň polynomu  $N$ , jak již bylo zmíněno. Při změně kombinace  $\alpha$  se, při porovnání četností jednotlivých částí obou histogramů, hodnoty statisticky významně neliší.

### 6.3 Testování odměňování Q-learningu

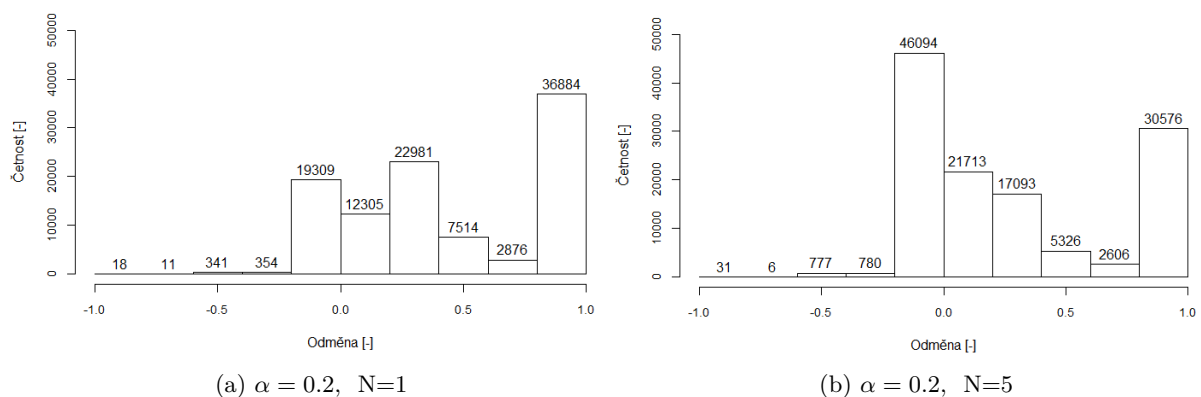
Odměna je hodnota, která je vypočítána vynásobením dílčích odměn za stav a zvolenou akci. Nabývá hodnot od -1 do 1, kdy interval -1 až 0 je vnímán negativně (penalizace) a interval 0 až 1 pozitivně (odměna).

Tabulka 11: Hodnoty kvartilů odměn/penalizace za akci a stav při jednotlivých kombinacích konstant

$\alpha$	N	Odměna [-]					
		MIN	1.Qv	2.Qv	MEAN	3.Qv	MAX
0,2	1	-1	0,1100	0,3300	0,4719	1	1
0,2	2	-1	0,0510	0,2200	0,4172	1	1
0,2	5	-1	-0,0500	0,1100	0,3076	0,77	1
0,5	1	-1	0,1100	0,3300	0,4718	1	1
0,5	2	-1	0,0310	0,2200	0,4164	1	1
0,5	5	-1	-0,0500	0,1100	0,3077	0,77	1

Všechny kombinace konstant mají shodnou hodnotu minima ( $MIN = -1$ ) a maxima ( $MAX = 1$ ). Odlišnosti začínají prvním kvantilem, který je u  $N=1$  a  $N=2$  kladný, u  $N=5$  záporný (nehledě na hodnotu rychlosti učení  $\alpha$ ). Tento fakt značí, že 25 % všech dat dostalo odměnu menší než je hodnota 1. kvartilu (viz tabulka č. 11). Důvodem relativně nízké četnosti záporných odměn (penalizací) je ten, že se systém učí rychle a penalizace už poté nejsou tolik zapotřebí.

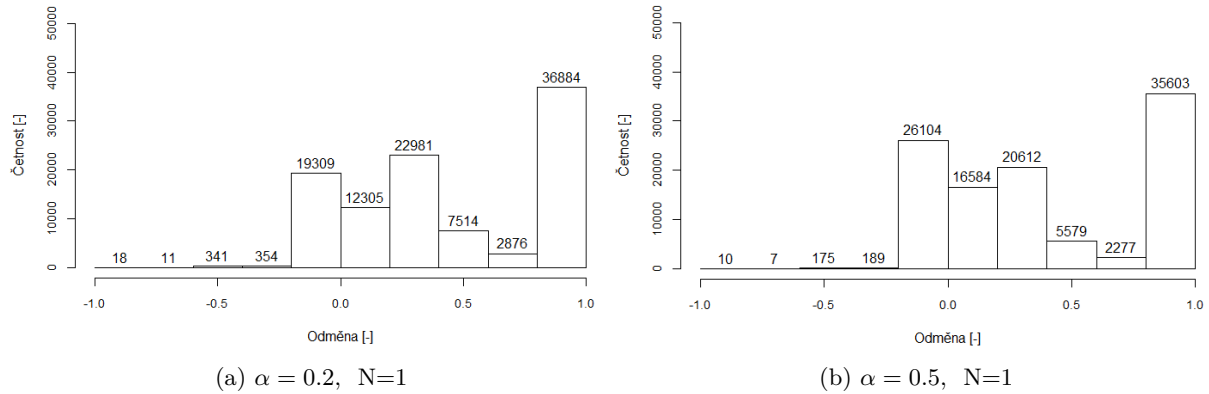
U mediánu (2. Qv) a průměru (MEAN) je očividná vzrůstající tendence hodnot odměny při snižujícím se  $N$ . Hodnota odměny  $\alpha = 0,2$   $N = 1$  je o 34,82 % vyšší než u  $\alpha = 0,2$   $N = 5$ . U  $\alpha = 0,5$   $N = 1$  je o 34,78 % vyšší než u  $\alpha = 0,5$   $N = 5$ . Z toho vychází, že obdobně jako u  $\varepsilon_{rel}$  nemá hodnota nastavení  $\alpha$  statisticky významný vliv na hodnotu odměn. U 3. kvartilu lze pozorovat odlišnost pouze u konstant  $\alpha = 0,5$   $N = 5$  a  $\alpha = 0,2$   $N = 5$ , což značí, že tyto kombinace měli horší volby akcí a predikci stavů než ostatní kombinace.



Obrázek 23: Histogram odměn v dané kombinaci konstant



Histogramy na obr. č. 24 potvrzují, že změna  $\alpha$  nemá statisticky významný vliv na hodnotu odměny. Oproti tomu grafy na obr. č. 23 potvrzují, že změna  $N$  má statisticky významně vliv na hodnotu odměny.



Obrázek 24: Histogram odměn v dané kombinaci konstant

## 6.4 Testování přesnosti predikce

$X_m$  jsou data, které jsou při započetí simulace dána. Jsou to data s informacemi o solární energii měřena s intervalem pět minut po dobu 5 let.  $X_{predict}$  jsou data, které vyprodukovala samotná simulace. Jsou to predikované hodnoty solární energie v daném čase.

Podstatou testování je porovnání, zda se predikované hodnoty  $X_{predict}$  statisticky významně liší/neliší od dat  $X_m$ . Obě proměnné byly testovány v simulaci s šesti různými kombinacemi expertních konstant a stupňů polynomu (viz tab. č. 5).

Statistické vyhodnocení bylo provedeno pomocí kvartilové analýzy, párového testování a výpočtu MSE, RMSE a SD.

### 6.4.1 Výsledky kvartilové analýzy

Minimum (MIN) je u všech kombinací konstant rovno 0 (viz tabulka č. 12). První změna je patrná u 1. kvartilu  $X_{predict}$   $\alpha = 0,2$   $N = 5$  (1.Qv=0,011) a  $X_{predict}$   $\alpha = 0,5$   $N = 5$  (1.Qv=0,068). Hodnoty maxima (MAX) jsou od 1144,20 do 1298,00. Důvodem jejich rozdílnosti je, že jsou statisticky vyhodnocovány pouze páry dat (nikoliv jednotlivé statistické soubory dat zvlášť jako celky).

U mediánu (2. Qv), aritmetického průměru (MEAN) a 3. kvartilu je patrná vzrůstající tendence hodnot se zvyšujícím se stupněm polynomu  $N$ , což dokazuje fakt, že hodnota aritmetického průměru  $X_m$   $\alpha = 0,2$   $N = 1$  je o 10,79 % vyšší než u  $X_m$   $\alpha = 0,2$   $N = 5$ . U  $X_m$   $\alpha = 0,5$   $N = 1$  je o 10,93 % vyšší než u  $X_m$   $\alpha = 0,5$   $N = 5$ . Z toho vychází, že obdobně jako u  $\varepsilon_{rel}$ , odměny a periody, nemá hodnota konstanty  $\alpha$  statisticky významný vliv.

Tabulka 12: Hodnoty kvartilů přejatých dat  $X_m$  a predikovaných dat  $X_{predict}$  při jednotlivých kombinacích konstant

$\alpha$ N			Energie [J]					
			MIN	1.Qv	2.Qv	MEAN	3.Qv	MAX
$X_m$	0,2	1	0	0	61,93	193,05	331,80	1298,00
	0,2	2	0	0	78,20	200,30	343,30	1298,00
	0,2	5	0	0	117,30	216,40	3674,00	1298,00
	0,5	1	0	0	61,66	193,01	331,90	1245,00
	0,5	2	0	0	79,50	201,20	346,10	1298,00
	0,5	5	0	0	117,10	216,70	367,80	1298,00
$X_{predict}$	0,2	1	0	0	61,93	193,05	331,80	1298,00
	0,2	2	0	0	80,10	200,34	350,50	1271,50
	0,2	5	0	0,011	122,54	216,37	377,88	1181,20
	0,5	1	0	0	61,66	193,00	331,90	1245,00
	0,5	2	0	0	82,08	201,17	352,80	1271,50
	0,5	5	0	0,068	122,48	216,69	379,32	1144,20

#### 6.4.2 Analýza rozdílů průměrů $X_m$ a $X_{predict}$

Při odečtení aritmetického průměru dat  $X_m$  a aritmetického průměru dat  $X_{predict}$  při daných konstantách (podle rovnice č. 13) je vypočítána hodnota rozdílu  $\Delta$ , viz tabulka č. 13.

$$\Delta = MEAN(X_m) - MEAN(X_{predict}) \quad (13)$$

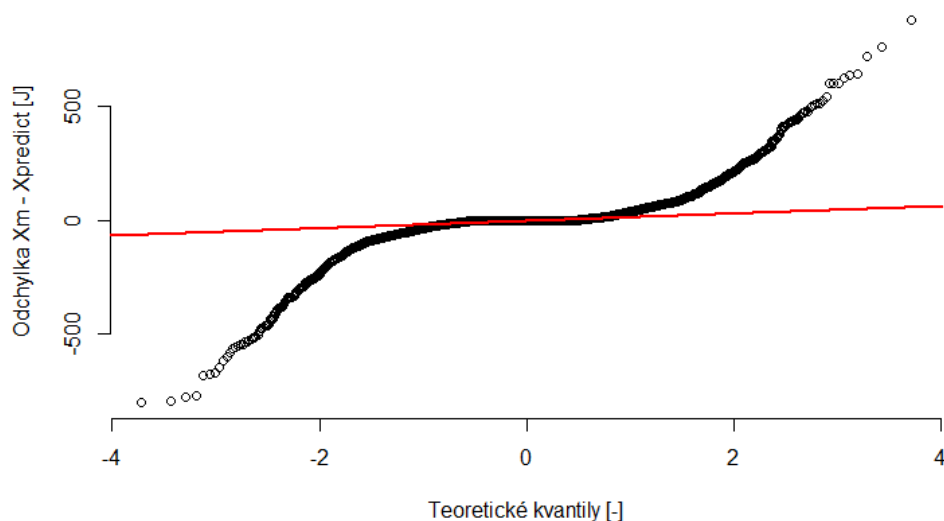
Se zvyšující se hodnotou stupně polynomu  $N$  se zvyšuje rozdíl  $\Delta$ . Například u  $X_m$   $\alpha = 0,5$   $N = 1$  je hodnota rozdílu o 53,33 % vyšší než u  $X_m$   $\alpha = 0,5$   $N = 5$ .

Tabulka 13: Hodnoty rozdílů průměru  $X_m$  a  $X_{predict}$  při jednotlivých kombinacích konstant

$\alpha$	N	MEAN $X_m$	MEAN $X_{predict}$	$\Delta = MEAN(X_m) - MEAN(X_{predict})$
0,2	1	193,05	193,05	0,0000
0,2	2	200,30	200,34	0,0006
0,2	5	216,40	216,37	0,0033
0,5	1	193,01	193,00	0,0014
0,5	2	201,20	201,17	0,0016
0,5	5	216,70	216,69	0,0030

### 6.4.3 Výsledky párového testování odchylek $X_m$ a $X_{predict}$

V tabulce č. 14 jsou výsledky párového porovnávání dvou datových souborů ( $X_m$  a  $X_{predict}$ ). Byly vypočítány odchylky (rozdíly) těchto datasetů. Datový soubor s odchylkami  $X_m$  a  $X_{predict}$ , při jednotlivých kombinacích, po využití Shapiro–Wilkova testu, nikdy nesplňoval podmínku normality (p-hodnota byla vždy menší než 5 %). Tuto skutečnost dokazuje i Q–Q graf na obr. č. 25, kde se body významně odlišují od proložené regresivní přímky. Proto další testování bylo provedeno pomocí mediánového Wilcoxonova testu.

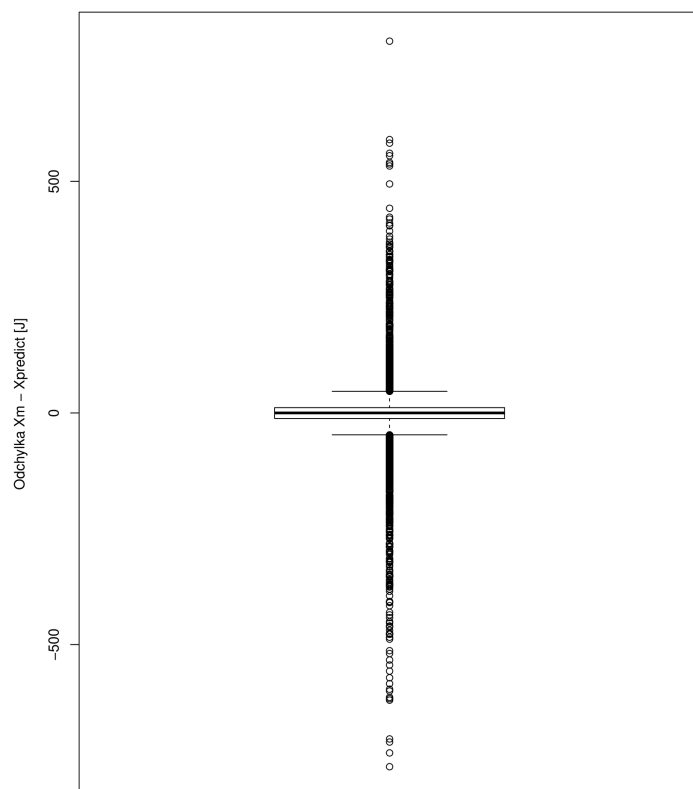


Obrázek 25: Q–Q graf odchylek  $X_m$  a  $X_{predict}$  při kombinacích konstant  $\alpha = 0,5$   $N = 1$

Výsledkem testu je p-hodnota, medián a 95% intervalový odhad. Jelikož p-hodnota byla vždy rovna hodnotě vyšší než 5 %, znamená to, že odchylky  $X_m$  a  $X_{predict}$  nejsou statisticky významně odlišné. Tuto skutečnost potvrzuje boxplot na obr. č. 26, na kterém je vidět, že odchylky jsou symetricky rozloženy kolem 0. Tzn, že součet kladných a záporných rozdílů se blíží k 0, ideálně je 0 (což potvrzuje také tabulka č. 13). Vysoká variabilita těchto odchylek je způsobená tím, že systém se učí postupně. Nepredikuje správné hodnoty hned od začátku simulace.

Interpretace výsledků uvedených v tabulce č. 14:

- Při kombinacích konstant  $\alpha = 0,2$   $N = 1$  je p-hodnota rovna 44,31 %. Medián odchylky se statisticky významně neliší od 0 a je roven 0,72 J. Můžeme jej očekávat v rozmezí intervalového odhadu od -1,15 do 2,70 J.
- Při kombinacích konstant  $\alpha = 0,2$   $N = 2$  je p-hodnota rovna 7,90 %. Medián odchylky se statisticky významně neliší od 0 a je roven 3,11 J. Můžeme jej očekávat v rozmezí intervalového odhadu od -3,65 do 1,95 J.



Obrázek 26: Boxplot odchylek  $X_m$  a  $X_{predict}$  při jednotlivých kombinacích konstant

Tabulka 14: Výsledky párového porovnávání odchylek  $X_m$  a  $X_{predict}$  při jednotlivých kombinacích konstant

$\alpha$	N	p-hodnota [%]	Medián	95% interval	
				od	do
0,2	1	44,31	0,72	-1,15	2,70
0,2	2	7,90	3,11	-0,34	6,80
0,2	5	37,46	-1,65	-5,63	2,12
0,5	1	24,02	1,70	-1,14	4,75
0,5	2	67,91	0,73	-2,72	4,33
0,5	5	71,19	0,73	-3,25	4,93

- Při kombinacích konstant  $\alpha = 0,2$   $N = 5$  je p-hodnota rovna 37,46 %. Medián odchylky se statisticky významně neliší od 0 a je roven -1,65 J. Můžeme jej očekávat v rozmezí intervalového odhadu od -5,63 do 2,12 J.
- Při kombinacích konstant  $\alpha = 0,5$   $N = 1$  je p-hodnota rovna 24,02 %. Medián odchylky se statisticky významně neliší od 0 a je roven 1,7 J. Můžeme jej očekávat v rozmezí inter-

valového odhadu od -1,14 do 4,75 J.

- Při kombinacích konstant  $\alpha = 0,5$   $N = 2$  je p-hodnota rovna 67,91 %. Medián odchylky se statisticky významně neliší od 0 a je roven 0,73 J. Můžeme jej očekávat v rozmezí intervalového odhadu od -2,72 do 4,33 J.
- Při kombinacích konstant  $\alpha = 0,5$   $N = 5$  je p-hodnota rovna 71,19 %. Medián odchylky se statisticky významně neliší od 0 a je roven 0,73 J. Můžeme jej očekávat v rozmezí intervalového odhadu od -3,25 do 4,93 J.

Záporné znaménko u mediánu znamená, že  $X_{predict}$  předbíhá (je větší než)  $X_m$ . Kladné znaménko značí, že  $X_{predict}$  se zpožďuje od (je menší než)  $X_m$ .

Jelikož z datového souboru odchylek  $X_m$  a  $X_{predict}$  je náhodně vybíráno pouze 5000 datových párů, výsledky párových testů, se při každém jiném náhodném výběru liší. Fakta, ale zůstávají stejná jako výše uvedená, což potvrzují grafická znázornění (např. test normality – Q–Q graf, mediánový Wilcoxonův test – boxplot). Z tohoto důvodu není možné porovnávat mezi sebou jednotlivé kombinace konstant pomocí párových testů, ale musí být využito výpočtů hodnot MSE, RMSE a SD (viz. tabulka č. 15). Párové testy slouží pouze k potvrzení/vyvrácení faktu, zdali se dva soubory dat od sebe statisticky liší či nikoliv.

#### 6.4.4 Analýza výsledků pomocí výpočtů MSE, RMSE a SD

Z tabulky č. 15 je patrné, že se zvyšující se hodnotou stupně polynomu  $N$  se zvyšuje hodnota MSE, RMSE a SD. To znamená, že vyšší stupeň polynomu zhoršuje kvalitu predikce. V porovnání s hodnotou MSE  $\alpha = 0,2$   $N = 1$  je hodnota MSE  $\alpha = 0,2$   $N = 5$  až o 30,39 %, s hodnotou RMSE  $\alpha = 0,2$   $N = 1$  je hodnota RMSE  $\alpha = 0,2$   $N = 5$  až o 16,56 %, s hodnotou SD  $\alpha = 0,2$   $N = 1$  je hodnota SD  $\alpha = 0,2$   $N = 5$  až o 16,61 % vyšší.

Tabulka 15: Výsledky MSE, RMSE a SD odchylek  $X_m$  a  $X_{predict}$  při jednotlivých kombinacích konstant

$\alpha$	N	MSE	RMSE	SD
0,2	1	7594,48	87,15	87,12
0,2	2	9315,90	96,52	96,54
0,2	5	10909,30	104,45	104,47
0,5	1	9534,60	97,65	97,65
0,5	2	9214,50	95,99	95,99
0,5	5	11644,90	107,90	107,90

Při zachování stupně polynomu  $N$  se se zvyšující hodnotou  $\alpha$  zvyšuje hodnota MSE, RMSE a SD (mimo  $N = 2$ , zde se snižuje). U  $\alpha = 0,5$   $N = 1$  je hodnota MSE o 20,35 % vyšší než u  $\alpha = 0,2$   $N = 1$ . U  $\alpha = 0,5$   $N = 5$  je hodnota MSE o 6,32 % vyšší než u  $\alpha = 0,2$   $N = 5$ .

Podle pravidla, čím je menší hodnota MSE, RMSE a SD, tím je přesnější predikce, by bylo pořadí kombinací konstant, od nejpřesnější po nejméně přesnou, následovné:

1.  $\alpha = 0,2$   $N = 1$ ,
2.  $\alpha = 0,5$   $N = 2$ ,
3.  $\alpha = 0,2$   $N = 2$ ,
4.  $\alpha = 0,5$   $N = 1$ ,
5.  $\alpha = 0,2$   $N = 5$ ,
6.  $\alpha = 0,5$   $N = 5$ .

#### 6.4.5 Analýza úbytku dat

Cílem simulace bylo měřit s co největší periodou, tzn. co nejméně často. Algoritmus proto vyprodukoval podstatně méně dat oproti přejatým datům z Kanady (viz tabulka č. ). I zde má na hodnotu ušetřených dat vliv stupeň polynomu  $N$ . Čím je větší stupeň polynomu, tím méně dat je ušetřeno. Při stejném  $N$  a změně  $\alpha$  je ušetřen stejný počet dat, s výjimkou  $\alpha = 0,2$ , kde je ovšem rozdíl roven desetině procenta.

Tabulka 16: Procentuální vyjádření úbytku dat při použití simulace oproti původním  $X_m$  při jednotlivých kombinacích konstant

$\alpha$	$N$	Ušetření dat [%]
0,2	1	95,96
0,2	2	95,75
0,2	5	95,08
0,5	1	95,95
0,5	2	95,75
0,5	5	95,08

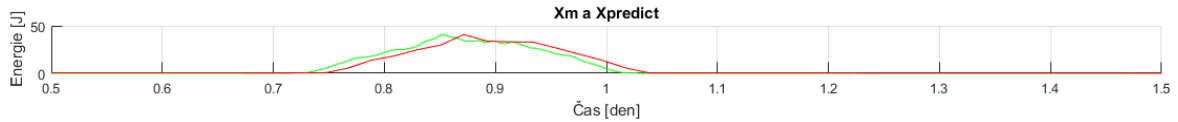
#### 6.4.6 Grafické výsledky simulace 1–3. den

V této podkapitole jsou uvedeny grafické výsledky simulace. Konkrétně porovnání dat  $X_m$  a  $X_{predict}$  ve stejných časových okamžicích. Průběhy energie jsou znázorněny od 1. do 3. dne (viz obr. č. 27). Zeleně zbarvenou křivkou je znázorněn  $X_m$ , červenou  $X_{predict}$ . Na grafu je vidět, jak se  $X_{predict}$  od počátku naučilo predikovat hodnoty energie. U každé kombinace je průběh rozdílný. U některých grafů lze pozorovat zpoždění  $X_{predict}$  za  $X_m$ , což potvrzují kladné hodnoty mediánu v tabulce č. 14. Naopak předbíhání udávají záporné hodnoty.



Obrázek 27: Graf srovnání průběhů, solární energie v závislosti na čase, predikce  $X_{predict}$  s přejatými daty  $X_m$  při kombinacích konstant  $\alpha = 0,2$   $N = 1$

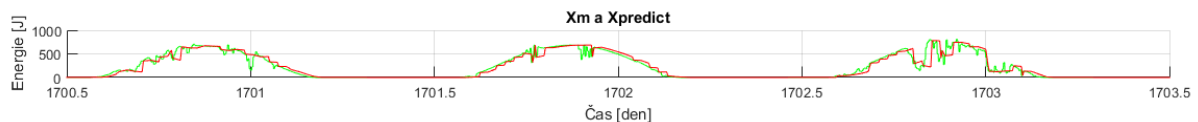
Konkrétně z obr. č. 27 je patrné, že pokud je nárůst/pokles energie pozvolný (1. den) je umí systém predikovat daleko lépe než při skocích (2. den). Přiblížení 1. dne měření (predikování) na obr. č. 28 poukazuje na zpoždění  $X_{predict}$  od  $X_m$ , což dokazuje kladná hodnota mediánu párového testování.



Obrázek 28: Graf srovnání průběhů, solární energie v závislosti na čase, predikce  $X_{predict}$  s přejatými daty  $X_m$  při kombinacích konstant  $\alpha = 0,2$   $N = 1$

#### 6.4.7 Grafické výsledky simulace 1701–1703. den

Na obr. č. 29 jsou porovnávány  $X_m$  a  $X_{predict}$  ve dnech od 1700 do 1703. Predikce je zde výrazně lepší v porovnání se dny 1–3. Predikce ve skokových úsecích je přesnější oproti začátkům simulace. Přiblížení dne č. 1701 je na obr. č. 30.



Obrázek 29: Graf srovnání průběhů, solární energie v závislosti na čase, predikce  $X_{predict}$  s přejatými daty  $X_m$  při kombinacích konstant  $\alpha = 0,2$   $N = 1$



Obrázek 30: Graf srovnání průběhů, solární energie v závislosti na čase, predikce  $X_{predict}$  s přejatými daty  $X_m$  při kombinacích konstant  $\alpha = 0,2$   $N = 1$

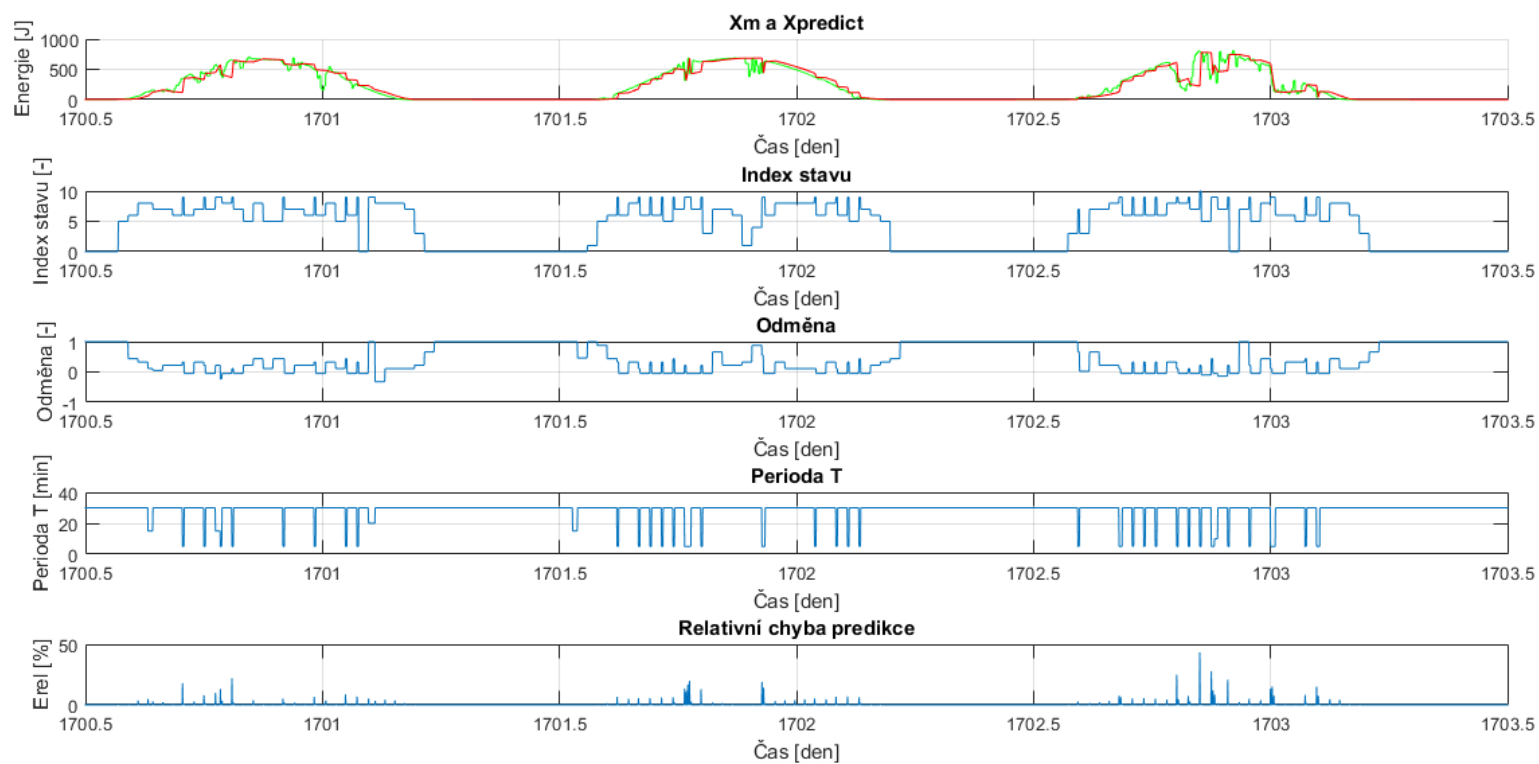
Na obrázku č. 31 je zobrazen průběh simulace od 1700. do 1703. dne. Mimo již zmíněný graf porovnání průběhů predikce  $X_{predict}$  s přejatými daty  $X_m$  obrázek navíc obsahuje grafy:

- Průběh indexu stavů v závislosti na čase – Graf ukazuje, který stav relativní chyby predikce  $\varepsilon_{rel}$  byl v daném časovém okamžiku použit. Index stavu 0 poukazuje na to, že byla hodnota  $\varepsilon_{rel}$  se pohybuje na intervalu od 0 do 0,02 %. Index stavu 1 poukazuje na to, že byla hodnota  $\varepsilon_{rel}$  se pohybuje na intervalu od 0,02 do 0,04 %. Takto analogicky se postupuje až do indexu stavu 10, který poukazuje na to, že hodnota  $\varepsilon_{rel}$  je na intervalu od 30 do 100 %. Jelikož  $\varepsilon_{rel}$  má v ideálním případě hodnotu 0 % (absolutní shoda predikované  $X_{predict}$  a přejaté hodnoty  $X_m$ ) je index stavu v okamžicích (např. 0 J), kdy můžeme na grafu

vidět přesnost predikce, v hodnotách 0. V opačném případě, při špatné predikci (např. den 1701), je hodnota indexu stavu od 5 do 10. Toto tvrzení potvrzuje spodní graf na téže obrázku.

- Průběh odměny v závislosti na čase – Graf ukazuje, že odměna se v těchto dnech téměř nedostane do svých možných záporných hodnot (důvodem je, že je systém již naučený predikovat). Četnost kladných odměn značí zpětnou vazbu systému, že predikce je úspěšná.
- Průběh periody v závislosti na čase – Tento graf znázorňuje změny periody, s jakou je změřena další hodnota. Je patrné, že největší četnost zastupuje perioda 30 min, která se nejčastěji střídá s periodou 5 min (tento fakt potvrzuje tabulka četností č. 7).
- Průběhů  $\varepsilon_{rel}$  v závislosti na čase – Na tomto grafu je vidět, jak se měnila relativní chyba predikce v průběhu času. Vyhodnocení viz druhý graf shora tohoto obrázku.





Obrázek 31: Grafy srovnání průběhů, solární energie v závislosti na čase, predikce  $X_{predict}$  s přejatými daty  $X_m$ ; Průběh indexu stavů v závislosti na čase; Průběh odměny v závislosti na čase; Průběh periody v závislosti na čase; Průběh  $\varepsilon_{rel}$  v závislosti na čase; Při kombinacích konstant  $\alpha = 0,2$   $N = 1$

## 6.5 Vyhodnocení výsledků testování

Z testování správné funkce simulace pomocí šesti kombinací expertních konstant a stupňů polynomu vychází následující.

Celkový počet naměřených dat ovlivňuje, které z možných period  $T$  jsou použity s nejvyšší četností. Zvolenou periodu přímo ovlivňuje volba stupně polynomu  $N$  a to takovým způsobem, že čím je větší  $N$ , tím je vyšší celkový počet naměřených dat a četnost periody 5 min (viz podkapitola č. 6.1). Četnost periody 30 min se naopak se zvyšující hodnotou stupně polynomu  $N$  snižuje. Nastavení rychlosti učení  $\alpha$  jej ovlivňuje v zanedbatelné míře. U všech kombinací byla s největší četností zastoupena perioda 30 min, což bylo cílem experimentu. Průměrná hodnota periody během měření byla u všech kombinací v intervalu od 20,31 do 24,75 min.

Relativní chyba predikce  $\varepsilon_{rel}$  se se zvyšujícím se stupněm polynomu  $N$ , zvyšuje. Z toho vychází, že nepřesnost predikce je u kombinací s vyššími stupni polynomu větší než u těch, jenž mají stupeň polynomu menší (viz podkapitola 6.2). Oproti tomu rozsah nepřesnosti predikce je u kombinací s nižšími stupni polynomu větší než u těch, jenž mají stupeň polynomu vyšší. Velikost rychlosti učení  $\alpha$  jeho, stejně jako periodu, ovlivňuje v zanedbatelné míře.

Odměnu taktéž ovlivňuje velikost stupně polynomu  $N$ . Se snižujícím se  $N$  se zvyšuje medián a průměrná hodnota odměny, tzn. odměna celkově (viz podkapitola č. 6.3). Zvolená rychlost učení  $\alpha$  na její velikost nemá statisticky významný vliv.

Při srovnání  $X_{predict}$  a  $X_m$  bylo dokázáno, že se zvyšující se hodnotu  $N$  se zvyšuje rozdíl průměrných hodnot  $X_{predict}$  a  $X_m$ , tzn. snižuje kvalita predikce. Při porovnávání odchylek  $X_{predict}$  od  $X_m$  bylo dokázáno, že, u všech kombinací konstant, mezi nimi není statisticky významný rozdíl. To znamená, že predikce dat byla u všech kombinací správná. Pro porovnání kvality predikce jednotlivých kombinací konstant mezi sebou bylo využito výpočtu MSE, RMSE a SD. Z výsledků těchto hodnot bylo potvrzeno, že vzrůstající hodnota stupně polynomu negativně ovlivňuje predikci (mimo  $N = 1$ ). Při zachování stejné hodnoty stupně polynomu a zvyšující se  $\alpha$  se zvyšuje hodnota MSE, RMSE a SD, tzn. zhoršuje kvalita predikce.

Významný rozdíl  $X_{predict}$  oproti  $X_m$  je v počtu vyprodukovaných dat.  $X_{predict}$  ušetřil, u všech kombinací, data v intervalu od 95,08 do 95,96 %. Konkrétně se jedná o přibližně 2,4 milionů ušetřených dat. I zde má na hodnotu ušetřených dat vliv stupeň polynomu  $N$  a to takový, že čím je menší stupeň polynomu, tím více dat je ušetřeno. Při stejném  $N$  a změně  $\alpha$  je ušetřen stejný počet dat, s výjimkou  $\alpha = 0,2$ , kde je rozdíl roven desetinně procenta.

## 7 Závěr

Energeticky nezávislá měřicí zařízení jsou hojně využívána v různých oblastech techniky - monitorování životního prostředí, varovné systémy před přírodními riziky nebo nositelná elektronika. Jejich princip tkví v tom, že nemají přístup k elektrizační soustavě a nejsou ovlivněny přítomností člověka. Cílem této práce bylo vytvořit algoritmus, který je schopen efektivně řídit měření dat v neznámém prostředí.

V teoretické části práce bylo popsáno zařízení jako takové (viz kapitola č. 2). Byly popsány oblasti, ve kterých nalézá využití. Zmíněny byly také jeho jednotlivé součásti. V další části práce byly popsány softcomputing metody, které se hodí k řízení energeticky nezávislých zařízení. Pro zpracování práce byla vybraná zpětnovazební metoda Q-learning.

V praktické části práce byl navržen Q-learning kontrolér (viz kapitola č. 4), který doplnil simulační software. Cílem navrženého algoritmu bylo, aby byl schopen předvídat následný vývoj podmínek prostředí, ve kterém se nachází a dle toho systém ovládal.

Byl složen z několika bloků, které daly za vznik výslednému kontroléru a následně jim byla doplněna simulace. Bloku *Buffer*, který funguje jako uložistiště dat, jenž slouží pro výpočet predikované hodnoty pomocí aproximace polynomem.

Dalším blokem je  $\varepsilon_{rel}$ , který vypočítává relativní chybu predikce, což je jeden ze dvou atributů sady pravidel učení Q-learningu. Samotný blok Q-learning slouží k učení systému predikovat následující periodu měření tak, aby nedošlo k ztrátě důležitých dat a zároveň, aby zbytečně nedocházelo k redundanci. Učení Q-learningu bylo definováno sadou pravidel, která říkala, za jaký stav/akce je odměna/penalizace. Stavem systému byl  $\varepsilon_{rel}$ , který nabýval hodnot od 0 do 100 %. Akce byly hodnoty periody  $T$ . Činnost Q-learningu byla navíc ovlivněna kombinací expertních konstant – rychlostí učení  $\alpha$ , diskontním faktorem  $\gamma$  a výběrem náhodných akcí  $\varepsilon$ .

Pro testování simulačního softwaru bylo použito šest kombinací expertních konstant a stupňů polynomu.

Data, ze kterých systém vycházel byla meteorologická data obsahující informaci o slunečním svitu. Byla to data z období 5 let, která byla sbírána při kontinuálním měření s periodou pět minut.

Metrikou vyhodnocení (viz kapitola č. 5) pro porovnání dvou datových souborů, výchozí hodnoty a predikované hodnoty, byly dvě metody. První byla použití párových testů a druhá výpočet hodnoty MSE, RMSE a SD. Pro další statistické vyhodnocení bylo použito výpočtů kvartilů, četností a graficky histogramů, boxplotů a Q-Q grafů.

Výsledky testování správné funkce simulace jsou shrnuty v kapitole č. 6. Vychází z nich, že velikost periody, odměny a relativní chyby predikce je závislá pouze na hodnotě stupně polynomu  $N$ .

U periody to je následovně – čím je vyšší stupeň polynomu, tím je vyšší četnost periody 5 min a nižší četnost periody 30 min. U všech kombinací byla s největší četností zastoupena perioda 30 min, což bylo cílem experimentu.

Relativní chyba predikce  $\varepsilon_{rel}$  se se zvyšujícím se stupněm polynomu  $N$ , zvyšuje. Z toho vychází, že nepřesnost predikce je u kombinací s vyššími stupni polynomu větší než u těch, jenž mají stupeň polynomu menší.

Odměnu taktéž ovlivňuje velikost stupně polynomu  $N$ . Se snižujícím se  $N$  se zvyšuje medián a průměrná hodnota odměny, tzn. odměna celkově. Závislost v této souvislosti na rychlosti učení  $\alpha$  nebyla u žádného atributu prokázána.

Co se týče testování rozdílu mezi  $X_{predict}$  a  $X_m$ , tak výsledky bylo potvrzeno, že vzrůstající hodnota stupně polynomu negativně ovlivňuje predikci (mimo  $N = 1$ ). Při zachování stejné hodnoty stupně polynomu a zvyšující se  $\alpha$  se zhoršuje kvalita predikce.

Při porovnávání odchylek  $X_{predict}$  od  $X_m$  bylo dokázáno, že, u všech kombinací konstant, mezi nimi není statisticky významný rozdíl. To znamená, že predikce dat byla u všech kombinací správná. Navíc došlo k ušetření dat, konkrétně se jedná o přibližně 2,4 milionů, a přesto predikce nevykazovala statisticky významný rozdíl. Z toho vychází, že algoritmus Q-learning kontroléru je schopen efektivně zaznamenávat data, aniž by docházelo k redundanci a ztrátě důležitých dat.

Dalším návrhem řešení práce by mohlo být testování větší variability kombinací expertních konstant a také změna nejen rychlosti učení  $\alpha$ , ale také diskontního faktor  $\gamma$  a výběru náhodných akcí  $\varepsilon$ . Mohl by být testován vliv kombinací akcí a stavů na přesnost predikce. Také by mohlo být rozčleněno učení na několik částí a mohlo by být sledováno s jakou rychlostí se systém učí a jaké jsou mezi jednotlivými částmi učení rozdíly. Dále by tento algoritmus mohl být otestován na jiných vstupních datech  $X_m$ .

## Literatura

- [1] NOVÁK, J. *Zpětnovazební učení pro řízení optimalizovaných vestavěných systémů*. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, 2019. Vedoucí práce Doc. Ing. Michal Prauzek, Ph.D.
- [2] WATKINS, Ch. J. CH. and P. DAYAN. *Q-learning. Machine learning*, 8(3-4): 279–292, 1992.
- [3] KULKARNI, R. V., A. FÖRSTER a G. KUMAR. *Computational Intelligence in Wireless Sensor Networks: A Survey.*, 2011.
- [4] Univerzitní systém Mendelu *Fuzzy logika[online]*. Dostupné z: [https://is.mendelu.cz/eknihovna/opory/zobraz\\_cast.pl?cast=21852](https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=21852)
- [5] VONDRÁK, I. *Neuronové sítě*. Ostrava: VŠB - Technická Univerzita Ostrava, 2009.
- [6] HAGAN, M. T., H. B. DEMUTH and M. H. BEALE. *Neural network design*. Boston: PWS Pub., c1996. ISBN 05-349-4332-2.
- [7] KROMER, P., M. PRAUZEK and P. MUSILEK. *Harvesting-aware control of wireless sensor nodes using fuzzy logic and differential evolution*. In 2014 11th Annual IEEE International Conference on Sensing, Communication, and Networking Workshops, SECON Workshops 2014, 2014.
- [8] KONEČNÝ, J., M. PRAUZEK, J. HLAVICA, J. NOVÁK a P. MUSILEK. *Simulation of a Daytime-based Q-learning Control Strategy for Environmental Harvesting WSN Nodes*. 2019.
- [9] MUSILEK, P., M. PRAUZEK, P. KROMER, J. RODWAY a T. BARTOŇ. *Intelligent Energy Management for Environmental Monitoring Systems*.
- [10] PRAUZEK, M., P. MUSILEK, and A. G. WATTS. *Fuzzy algorithm for intelligent wireless sensors with solar harvesting*. 2014.
- [11] Glossary. *Fuzzy Logic Toolbox.*, 2002.
- [12] BENNETT, P. *The why, where and what of low-power SoC design*. EE Times, 2004.
- [13] TRONG, N. L., A. PEGATOQUET, O. BERDER, O. SENTIEYS and A. CARER. *Energy-neutral design framework for supercapacitor-based autonomous wireless sensor networks*. ACM Journal on Emerging Technologies in Computing Systems (JETC), 2015.
- [14] PRAUZEK, M., J. KONEČNÝ, M. BOROŮVÁ, K. JANOŠOVÁ, J. HLAVICA, P. MUSILEK. *Energy harvesting sources, storage devices and system topologies for environmental wireless sensor networks: A review*. VŠB-TUO, 2018.

- [15] ROUSE, M. *What is a Microcontroller and How Does it Work?* IoT Agenda, 2012.
- [16] TEWARI, D. *Introduction of Microprocessor*. GeeksforGeeks, 2018.
- [17] Critical Link. *Field Programmable Gate Arrays*, 2019.
- [18] RAJEWSKI, J. *What is an FPGA?* Alchitry, 2018.
- [19] LITSCHMANNOVÁ, M. *Přednášky (BioSTA) VŠB-TUO: Katedra aplikované matematiky*, 2012
- [20] SCHOVÁNEK, P. a HAVRÁNEK, V. *Chyby a nejistoty měření -doplňující text k laboratornímu cvičení ČVUT*
- [21] PRAUZEK, M. *Metody řízení pro energeticky nezávislé vestavěné měřicí systémy* 2017. Habilitační práce. VŠB-TUO, FEI.
- [22] HAVLÍK, J. *Hodnocení kvality odhadu stavu stochastických systémů* Plzeň, 2014. Diplomová práce. Západočeská univerzita v Plzni.
- [23] Veterinární a farmaceutická univerzita Brno: Statistika a INFORMATIKA [online]. *Wilcoxon* Dostupné z: <https://cit.vfu.cz/statpotr/POTR/Teorie/Predn4/Wilcoxon.htm>
- [24] Matematika.cz [online]. *Směrodatná odchylka — Matematika.cz*. Vydavatelství Nová média. Dostupné z: <https://matematika.cz/smerodatna-odchylka>
- [25] HABALA, P. *Math Tutor - Functions - Theory - Elementary Functions [online]* 2001. Dostupné z: <http://math.feld.cvut.cz/mt/txtb/4/txc3ba4b.htm>
- [26] Aproximace funkcí [online] *Aproximace funkcí*. 20.10.2013. Dostupné z: [http : //home.zcu.cz/ danek/DATA/WWW\\_STRANKY/soubory/aproximace\\_funkci.pdf](http://home.zcu.cz/danek/DATA/WWW_STRANKY/soubory/aproximace_funkci.pdf)
- [27] LIKEŠ, J. a J. MACHEK. *Matematická statistika* SNTL Praha, 1988.
- [28] Simplefpga.blogspot.com *Learn Verilog by Example: FIFO(First In First Out) Buffer in Verilog*. In: Simplefpga.blogspot.com [online]. Dostupné z: <http://simplefpga.blogspot.com/2012/12/fifofirst-in-first-out-buffer-in-verilog.html>
- [29] KONEČNÝ, J., M. PRAUZEK, J. HLAVICA, J. NOVÁK a P. MUSILEK. *Simulation of a Daytime-based Q-learning Control Strategy for Environmental Harvesting WSN Nodes [online]*, 2019.
- [30] Výpočet kvartilů Q1, Q2, Q3 [online]. *Výpočet kvartilů Q1, Q2, Q3 (statistika)*. Dostupné z: <https://www.hackmath.net/cz/kalkulacka/vypocet-kvartilu-q1-q2-q3>
- [31] Matematika.cz [online]. *Základy statistiky — Matematika.cz*. Dostupné z: <https://matematika.cz/zaklady-statistiky>

- [32] Artificial Intelligence - foundations of computational agents [online]. *Q-learning*. 2010. Dostupné z: [https : //artint.info/html/ArtInt265.html](https://artint.info/html/ArtInt265.html)
- [33] Towards data science [online]. *Simple Reinforcement Learning: Q-learning - Towards Data Science*. Dostupné z: <https://towardsdatascience.com/simple-reinforcement-learning-q-learning-fcddc4b6fe56>
- [34] PALMIERI, L. *Reinforcement Learning: a comprehensive introduction*, 2018. Dostupné z: <https://www.lpalmieri.com/posts/rl-introduction-00/>.
- [35] GHAED, M. H. et al. *Circuits for a Cubic-Millimeter Energy-Autonomous Wireless Intra-ocular Pressure Monitor* in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 60, no. 12, pp. 3152-3162, Dec. 2013.
- [36] NUNEZ, G., C. MANJAKKAL, L. DAHIYA *Energy autonomous electronic skin*. *npj Flex Electron* 3, 1. 2019. Dostupné z: <https://doi.org/10.1038/s41528-018-0045-x>
- [37] YUEN, J.D., A. BAINGANE and Q. HASAN et al. *A Fully-Flexible Solution-Processed Autonomous Glucose Indicator*. *Sci Rep* 9, 6931 (2019). Dostupné z: <https://doi.org/10.1038/s41598-019-43425-x>